

Kostas Daniilidis  
Petros Maragos  
Nikos Paragios (Eds.)

LNCS 6311

# Computer Vision – ECCV 2010

11th European Conference on Computer Vision  
Heraklion, Crete, Greece, September 2010  
Proceedings, Part I

1  
Part I



 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*

Kostas Daniilidis Petros Maragos  
Nikos Paragios (Eds.)

# Computer Vision – ECCV 2010

11th European Conference on Computer Vision  
Heraklion, Crete, Greece, September 5-11, 2010  
Proceedings, Part I

Volume Editors

Kostas Daniilidis  
GRASP Laboratory, University of Pennsylvania  
3330 Walnut Street, Philadelphia, PA 19104, USA  
E-mail: kostas@cis.upenn.edu

Petros Maragos  
National Technical University of Athens  
School of Electrical and Computer Engineering  
15773 Athens, Greece  
E-mail: maragos@cs.ntua.gr

Nikos Paragios  
Ecole Centrale de Paris  
Department of Applied Mathematics  
Grande Voie des Vignes, 92295 Chatenay-Malabry, France  
E-mail: nikos.paragios@ecp.fr

Library of Congress Control Number: 2010933243

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

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

ISSN 0302-9743  
ISBN-10 3-642-15548-0 Springer Berlin Heidelberg New York  
ISBN-13 978-3-642-15548-2 Springer Berlin Heidelberg New York

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

springer.com

© Springer-Verlag Berlin Heidelberg 2010  
Printed in Germany

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

## Preface

The 2010 edition of the European Conference on Computer Vision was held in Heraklion, Crete. The call for papers attracted an absolute record of 1,174 submissions. We describe here the selection of the accepted papers:

- Thirty-eight area chairs were selected coming from Europe (18), USA and Canada (16), and Asia (4). Their selection was based on the following criteria: (1) Researchers who had served at least two times as Area Chairs within the past two years at major vision conferences were excluded; (2) Researchers who served as Area Chairs at the 2010 Computer Vision and Pattern Recognition were also excluded (exception: ECCV 2012 Program Chairs); (3) Minimization of overlap introduced by Area Chairs being former student and advisors; (4) 20% of the Area Chairs had never served before in a major conference; (5) The Area Chair selection process made all possible efforts to achieve a reasonable geographic distribution between countries, thematic areas and trends in computer vision.
- Each Area Chair was assigned by the Program Chairs between 28–32 papers. Based on paper content, the Area Chair recommended up to seven potential reviewers per paper. Such assignment was made using all reviewers in the database including the conflicting ones. The Program Chairs manually entered the missing conflict domains of approximately 300 reviewers. Based on the recommendation of the Area Chairs, three reviewers were selected per paper (with at least one being of the top three suggestions), with 99.7% being the recommendations of the Area Chairs. When this was not possible, senior reviewers were assigned to these papers by the Program Chairs, with the consent of the Area Chairs. Upon completion of this process there were 653 active reviewers in the system.
- Each reviewer got a maximum load of eight reviews—in a few cases we had nine papers when re-assignments were made manually because of hidden conflicts. Upon the completion of the reviews deadline, 38 reviews were missing. The Program Chairs proceeded with fast re-assignment of these papers to senior reviewers. Prior to the deadline of submitting the rebuttal by

the authors, all papers had three reviews. The distribution of the reviews was the following: 100 papers with an average score of weak accept and higher, 125 papers with an average score toward weak accept, 425 papers with an average score around borderline.

- For papers with strong consensus among reviewers, we introduced a procedure to handle potential overwriting of the recommendation by the Area Chair. In particular for all papers with weak accept and higher or with weak reject and lower, the Area Chair should have sought for an additional reviewer prior to the Area Chair meeting. The decision of the paper could have been changed if the additional reviewer was supporting the recommendation of the Area Chair, and the Area Chair was able to convince his/her group of Area Chairs of that decision.
- The discussion phase between the Area Chair and the reviewers was initiated once the review became available. The Area Chairs had to provide their identity to the reviewers. The discussion remained open until the Area Chair meeting that was held in Paris, June 5–6. Each Area Chair was paired to a buddy and the decisions for all papers were made jointly, or when needed using the opinion of other Area Chairs. The pairing was done considering conflicts, thematic proximity, and when possible geographic diversity. The Area Chairs were responsible for taking decisions on their papers. Prior to the Area Chair meeting, 92% of the consolidation reports and the decision suggestions had been made by the Area Chairs. These recommendations were used as a basis for the final decisions.
- Orals were discussed in groups of Area Chairs. Four groups were formed, with no direct conflict between paper conflicts and the participating Area Chairs. The Area Chair recommending a paper had to present the paper to the whole group and explain why such a contribution is worth being published as an oral. In most of the cases consensus was reached in the group, while in the cases where discrepancies existed between the Area Chairs' views, the decision was taken according to the majority of opinions.
- The final outcome of the Area Chair meeting, was 38 papers accepted for an oral presentation and 284 for poster. The percentage ratios of submissions/ acceptance per area are the following:

Thematic area	# submitted	% over submitted	# accepted	% over accepted	% acceptance in area
Object and Scene Recognition	192	16.4%	66	20.3%	34.4%
Segmentation and Grouping	129	11.0%	28	8.6%	21.7%
Face, Gesture, Biometrics	125	10.6%	32	9.8%	25.6%
Motion and Tracking	119	10.1%	27	8.3%	22.7%
Statistical Models and Visual Learning	101	8.6%	30	9.2%	29.7%
Matching, Registration, Alignment	90	7.7%	21	6.5%	23.3%
Computational Imaging	74	6.3%	24	7.4%	32.4%
Multi-view Geometry	67	5.7%	24	7.4%	35.8%
Image Features	66	5.6%	17	5.2%	25.8%
Video and Event Characterization	62	5.3%	14	4.3%	22.6%
Shape Representation and Recognition	48	4.1%	19	5.8%	39.6%
Stereo	38	3.2%	4	1.2%	10.5%
Reflectance, Illumination, Color	37	3.2%	14	4.3%	37.8%
Medical Image Analysis	26	2.2%	5	1.5%	19.2%

- We received 14 complaints/reconsideration requests. All of them were sent to the Area Chairs who handled the papers. Based on the reviewers' arguments and the reaction of the Area Chair, three papers were accepted—as posters—on top of the 322 at the Area Chair meeting, bringing the total number of accepted papers to 325 or **27.6%**. The selection rate for the 38 orals was **3.2%**. The acceptance rate for the papers submitted by the group of Area Chairs was 39%.
- Award nominations were proposed by the Area and Program Chairs based on the reviews and the consolidation report. An external award committee was formed comprising David Fleet, Luc Van Gool, Bernt Schiele, Alan Yuille, Ramin Zabih. Additional reviews were considered for the nominated papers and the decision on the paper awards was made by the award committee. We thank the Area Chairs, Reviewers, Award Committee Members, and the General Chairs for their hard work and we gratefully acknowledge Microsoft Research for accommodating the ECCV needs by generously providing the CMT Conference Management Toolkit. We hope you enjoy the proceedings.

# Organization

## General Chairs

Argyros, Antonis	University of Crete/FORTH, Greece
Trahanias, Panos	University of Crete/FORTH, Greece
Tziritas, George	University of Crete, Greece

## Program Chairs

Daniilidis, Kostas	University of Pennsylvania, USA
Maragos, Petros	National Technical University of Athens, Greece
Paragios, Nikos	Ecole Centrale de Paris/INRIA Saclay île-de-France, France

## Workshops Chair

Kutulakos, Kyros	University of Toronto, Canada
------------------	-------------------------------

## Tutorials Chair

Lourakis, Manolis	FORTH, Greece
-------------------	---------------

## Demonstrations Chair

Kakadiaris, Ioannis	University of Houston, USA
---------------------	----------------------------

## Industrial Chair

Pavlidis, Ioannis	University of Houston, USA
-------------------	----------------------------

## Travel Grants Chair

Komodakis, Nikos	University of Crete, Greece
------------------	-----------------------------



## Area Chairs

Bach, Francis	INRIA Paris - Rocquencourt, France
Belongie, Serge	University of California-San Diego, USA
Bischof, Horst	Graz University of Technology, Austria
Black, Michael	Brown University, USA
Boyer, Edmond	INRIA Grenoble - Rhône-Alpes, France
Cootes, Tim	University of Manchester, UK
Dana, Kristin	Rutgers University, USA
Davis, Larry	University of Maryland, USA
Efros, Alyosha	Carnegie Mellon University, USA
Fermuller, Cornelia	University of Maryland, USA
Fitzgibbon, Andrew	Microsoft Research, Cambridge, UK
Jepson, Alan	University of Toronto, Canada
Kahl, Fredrik	Lund University, Sweden
Keriven, Renaud	Ecole des Ponts-ParisTech, France
Kimmel, Ron	Technion Institute of Technology, Ireland
Kolmogorov, Vladimir	University College of London, UK
Lepetit, Vincent	Ecole Polytechnique Federale de Lausanne, Switzerland
Matas, Jiri	Czech Technical University, Prague, Czech Republic
Metaxas, Dimitris	Rutgers University, USA
Navab, Nassir	Technical University of Munich, Germany
Nister, David	Microsoft Research, Redmont, USA
Perez, Patrick	THOMSON Research, France
Perona, Pietro	Caltech University, USA
Ramesh, Visvanathan	Siemens Corporate Research, USA
Raskar, Ramesh	Massachusetts Institute of Technology, USA
Samaras, Dimitris	State University of New York - Stony Brook, USA
Sato, Yoichi	University of Tokyo, Japan
Schmid, Cordelia	INRIA Grenoble - Rhône-Alpes, France
Schnoerr, Christoph	University of Heidelberg, Germany
Sebe, Nicu	University of Trento, Italy
Szeliski, Richard	Microsoft Research, Redmont, USA
Taskar, Ben	University of Pennsylvania, USA
Torr, Phil	Oxford Brookes University, UK
Torralba, Antonio	Massachusetts Institute of Technology, USA
Tuytelaars, Tinne	Katholieke Universiteit Leuven, Belgium
Weickert, Joachim	Saarland University, Germany
Weinshall, Daphna	Hebrew University of Jerusalem, Israel
Weiss, Yair	Hebrew University of Jerusalem, Israel

## Conference Board

Horst Bischof	Graz University of Technology, Austria
Hans Burkhardt	University of Freiburg, Germany
Bernard Buxton	University College London, UK
Roberto Cipolla	University of Cambridge, UK
Jan-Olof Eklundh	Royal Institute of Technology, Sweden
Olivier Faugeras	INRIA, Sophia Antipolis, France
David Forsyth	University of Illinois, USA
Anders Heyden	Lund University, Sweden
Ales Leonardis	University of Ljubljana, Slovenia
Bernd Neumann	University of Hamburg, Germany
Mads Nielsen	IT University of Copenhagen, Denmark
Tomas Pajdla	CTU Prague, Czech Republic
Jean Ponce	Ecole Normale Superieure, France
Giulio Sandini	University of Genoa, Italy
Philip Torr	Oxford Brookes University, UK
David Vernon	Trinity College, Ireland
Andrew Zisserman	University of Oxford, UK

## Reviewers

Abd-Almageed, Wael	Bahlmann, Claus	Bougleux, Sebastien
Agapito, Lourdes	Baker, Simon	Boult, Terrance
Agarwal, Sameer	Ballan, Luca	Boureau, Y-Lan
Aggarwal, Gaurav	Barbu, Adrian	Bowden, Richard
Ahlberg, Juergen	Barnes, Nick	Boykov, Yuri
Ahonen, Timo	Barreto, Joao	Bradski, Gary
Ai, Haizhou	Bartlett, Marian	Bregler, Christoph
Alahari, Karteek	Bartoli, Adrien	Bremond, Francois
Aleman-Flores, Miguel	Batra, Dhruv	Bronstein, Alex
Aloimonos, Yiannis	Baust, Maximilian	Bronstein, Michael
Amberg, Brian	Beardsley, Paul	Brown, Matthew
Andreetto, Marco	Behera, Ardhendu	Brown, Michael
Angelopoulou, Elli	Beleznai, Csaba	Brox, Thomas
Ansar, Adnan	Ben-ezra, Moshe	Brubaker, Marcus
Arbel, Tal	Berg, Alexander	Bruckstein, Freddy
Arbelaez, Pablo	Berg, Tamara	Bruhn, Andres
Astroem, Kalle	Betke, Margrit	Buisson, Olivier
Athitsos, Vassilis	Bileschi, Stan	Burkhardt, Hans
August, Jonas	Birchfield, Stan	Burschka, Darius
Avraham, Tamar	Biswas, Soma	Caetano, Tiberio
Azzabou, Noura	Blanz, Volker	Cai, Deng
Babenko, Boris	Blaschko, Matthew	Calway, Andrew
Bagdanov, Andrew	Bobick, Aaron	Cappelli, Raffaele

Caputo, Barbara	Domke, Justin	Fua, Pascal
Carreira-Perpinan, Miguel	Donoser, Michael	Fuchs, Martin
Caselles, Vincent	Doretto, Gianfranco	Furukawa, Yasutaka
Cavallaro, Andrea	Douze, Matthijs	Fusiello, Andrea
Cham, Tat-Jen	Draper, Bruce	Gall, Juergen
Chandraker, Manmohan	Drbohlav, Ondrej	Gallagher, Andrew
Chandran, Sharat	Duan, Qi	Gao, Xiang
Chetverikov, Dmitry	Duchenne, Olivier	Gatica-Perez, Daniel
Chiu, Han-Pang	Duric, Zoran	Gee, James
Cho, Taeg Sang	Duygulu-Sahin, Pinar	Gehler, Peter
Chuang, Yung-Yu	Eklundh, Jan-Olof	Genc, Yakup
Chung, Albert C. S.	Elder, James	Georgescu, Bogdan
Chung, Moo	Elgammal, Ahmed	Geusebroek, Jan-Mark
Clark, James	Epshtein, Boris	Gevers, Theo
Cohen, Isaac	Eriksson, Anders	Geyer, Christopher
Collins, Robert	Espuny, Ferran	Ghosh, Abhijeet
Colombo, Carlo	Essa, Irfan	Glocker, Ben
Cord, Matthieu	Farhadi, Ali	Goecke, Roland
Corso, Jason	Farrell, Ryan	Goedeme, Toon
Costen, Nicholas	Favaro, Paolo	Goldberger, Jacob
Cour, Timothee	Fehr, Janis	Goldenstein, Siome
Crandall, David	Fei-Fei, Li	Goldluecke, Bastian
Cremers, Daniel	Felsberg, Michael	Gomes, Ryan
Criminisi, Antonio	Ferencz, Andras	Gong, Sean
Crowley, James	Fergus, Rob	Gorelick, Lena
Cui, Jinshi	Feris, Rogerio	Gould, Stephen
Cula, Oana	Ferrari, Vittorio	Grabner, Helmut
Dalalyan, Arnak	Ferryman, James	Grady, Leo
Darbon, Jerome	Fidler, Sanja	Grau, Oliver
Davis, James	Finlayson, Graham	Grauman, Kristen
Davison, Andrew	Fisher, Robert	Gross, Ralph
de Bruijne, Marleen	Flach, Boris	Grossmann, Etienne
De la Torre, Fernando	Fleet, David	Gruber, Amit
Dedeoglu, Goksel	Fletcher, Tom	Gulshan, Varun
Delong, Andrew	Florack, Luc	Guo, Guodong
Demirci, Stefanie	Flynn, Patrick	Gupta, Abhinav
Demirdjian, David	Foerstner, Wolfgang	Gupta, Mohit
Denzler, Joachim	Foroosh, Hassan	Habbecke, Martin
Deselaers, Thomas	Forssen, Per-Erik	Hager, Gregory
Dhome, Michel	Fowlkes, Charless	Hamid, Raffay
Dick, Anthony	Frahm, Jan-Michael	Han, Bohyung
Dickinson, Sven	Fraundorfer, Friedrich	Han, Tony
Divakaran, Ajay	Freeman, William	Hanbury, Allan
Dollar, Piotr	Frey, Brendan	Hancock, Edwin
	Fritz, Mario	Hasinoff, Samuel

Hassner, Tal	Kamarainen,	Larlus, Diane
Haussecker, Horst	Joni-Kristian	Latecki, Longin Jan
Hays, James	Kamberov, George	Lazebnik, Svetlana
He, Xuming	Kamberova, Gerda	Lee, ChanSu
Heas, Patrick	Kambhamettu, Chandra	Lee, Honglak
Hebert, Martial	Kanatani, Kenichi	Lee, Kyoung Mu
Heibel, T. Hauke	Kanaujia, Atul	Lee, Sang-Wook
Heidrich, Wolfgang	Kang, Sing Bing	Leibe, Bastian
Hernandez, Carlos	Kappes, Jörg	Leichter, Ido
Hilton, Adrian	Kavukcuoglu, Koray	Leistner, Christian
Hinterstoisser, Stefan	Kawakami, Rei	Lellmann, Jan
Hlavac, Vaclav	Ke, Qifa	Lempitsky, Victor
Hoiem, Derek	Kemelmacher, Ira	Lenzen, Frank
Hoogs, Anthony	Khamene, Ali	Leonardis, Ales
Hornegger, Joachim	Khan, Saad	Leung, Thomas
Hua, Gang	Kikinis, Ron	Levin, Anat
Huang, Rui	Kim, Seon Joo	Li, Chunming
Huang, Xiaolei	Kimia, Benjamin	Li, Gang
Huber, Daniel	Kittler, Josef	Li, Hongdong
Hudelot, Celine	Koch, Reinhard	Li, Hongsheng
Hussein, Mohamed	Koeser, Kevin	Li, Li-Jia
Huttenlocher, Dan	Kohli, Pushmeet	Li, Rui
Ihler, Alex	Kokiopoulou, Efi	Li, Ruonan
Ilic, Slobodan	Kokkinos, Iasonas	Li, Stan
Irschara, Arnold	Kolev, Kalin	Li, Yi
Ishikawa, Hiroshi	Komodakis, Nikos	Li, Yunpeng
Isler, Volkan	Konolige, Kurt	Liefeng, Bo
Jain, Prateek	Koschan, Andreas	Lim, Jongwoo
Jain, Viren	Kukelova, Zuzana	Lin, Stephen
Jamie Shotton, Jamie	Kulis, Brian	Lin, Zhe
Jegou, Herve	Kumar, M. Pawan	Ling, Haibin
Jenatton, Rodolphe	Kumar, Sanjiv	Little, Jim
Jermyn, Ian	Kuthirummal, Sujit	Liu, Ce
Ji, Hui	Kutulakos, Kyros	Liu, Jingen
Ji, Qiang	Kweon, In So	Liu, Qingshan
Jia, Jiaya	Ladicky, Lubor	Liu, Tyng-Luh
Jin, Hailin	Lai, Shang-Hong	Liu, Xiaoming
Jogan, Matjaz	Lalonde, Jean-Francois	Liu, Yanxi
Johnson, Micah	Lampert, Christoph	Liu, Yazhou
Joshi, Neel	Landon, George	Liu, Zicheng
Juan, Olivier	Langer, Michael	Lourakis, Manolis
Jurie, Frederic	Langs, Georg	Lovell, Brian
Kakadiaris, Ioannis	Lanman, Douglas	Lu, Le
Kale, Amit	Laptev, Ivan	Lucey, Simon

Luo, Jiebo	Mukaigawa, Yasuhiro	Peleg, Shmuel
Lyu, Siwei	Mulligan, Jane	Perera, A.G. Amitha
Ma, Xiaoxu	Munich, Mario	Perronnin, Florent
Mairal, Julien	Murino, Vittorio	Petrou, Maria
Maire, Michael	Namboodiri, Vinay	Petrovic, Vladimir
Maji, Subhransu	Narasimhan, Srinivasa	Peursum, Patrick
Maki, Atsuto	Narayanan, P.J.	Philbin, James
Makris, Dimitrios	Naroditsky, Oleg	Piater, Justus
Malisiewicz, Tomasz	Neumann, Jan	Pietikainen, Matti
Mallick, Satya	Nevatia, Ram	Pinz, Axel
Manduchi, Roberto	Nicolls, Fred	Pless, Robert
Manmatha, R.	Niebles, Juan Carlos	Pock, Thomas
Marchand, Eric	Nielsen, Mads	Poh, Norman
Marcialis, Gian	Nishino, Ko	Pollefeys, Marc
Marks, Tim	Nixon, Mark	Ponce, Jean
Marszalek, Marcin	Nowozin, Sebastian	Pons, Jean-Philippe
Martinec, Daniel	O'donnell, Thomas	Potetz, Brian
Martinez, Aleix	Obozinski, Guillaume	Prabhakar, Salil
Matei, Bogdan	Odobez, Jean-Marc	Qian, Gang
Mateus, Diana	Odone, Francesca	Quattoni, Ariadna
Matsushita, Yasuyuki	Ofek, Eyal	Radeva, Petia
Matthews, Iain	Ogale, Abhijit	Radke, Richard
Maxwell, Bruce	Okabe, Takahiro	Rakotomamonjy, Alain
Maybank, Stephen	Okatani, Takayuki	Ramanan, Deva
Mayer, Helmut	Okuma, Kenji	Ramanathan, Narayanan
McCloskey, Scott	Olson, Clark	Ranzato, Marc'Aurelio
McKenna, Stephen	Olsson, Carl	Raviv, Dan
Medioni, Gerard	Ommer, Bjorn	Reid, Ian
Meer, Peter	Osadchy, Margarita	Reitmayr, Gerhard
Mei, Christopher	Overgaard, Niels	Ren, Xiaofeng
Michael, Nicholas	Christian	Rittscher, Jens
Micusik, Branislav	Ozuysal, Mustafa	Rogez, Gregory
Minh, Nguyen	Pajdla, Tomas	Rosales, Romer
Mirmehdi, Majid	Panagopoulos,	Rosenberg, Charles
Mittal, Anurag	Alexandros	Rosenhahn, Bodo
Miyazaki, Daisuke	Pandharkar, Rohit	Rosman, Guy
Monasse, Pascal	Pankanti, Sharath	Ross, Arun
Mordohai, Philippos	Pantic, Maja	Roth, Peter
Moreno-Noguer,	Papadopoulo, Theo	Rother, Carsten
Francesc	Parameswaran, Vasu	Rothganger, Fred
Mori, Greg	Parikh, Devi	Rougon, Nicolas
Morimoto, Carlos	Paris, Sylvain	Roy, Sebastien
Morse, Bryan	Patow, Gustavo	Rueckert, Daniel
Moses, Yael	Patras, Ioannis	Ruether, Matthias
Mueller, Henning	Pavlovic, Vladimir	Russell, Bryan

- Russell, Christopher  
 Sahbi, Hichem  
 Stiefelwagen, Rainer  
 Saad, Ali  
 Saffari, Amir  
 Salgian, Garbis  
 Salzmann, Mathieu  
 Sangineto, Enver  
 Sankaranarayanan,  
     Aswin  
 Sapiro, Guillermo  
 Sara, Radim  
 Sato, Imari  
 Savarese, Silvio  
 Savchynskyy, Bogdan  
 Sawhney, Harpreet  
 Scharr, Hanno  
 Scharstein, Daniel  
 Schellewald, Christian  
 Schiele, Bernt  
 Schindler, Grant  
 Schindler, Konrad  
 Schlesinger, Dmitrij  
 Schoenemann, Thomas  
 Schroff, Florian  
 Schubert, Falk  
 Schultz, Thomas  
 Se, Stephen  
 Seidel, Hans-Peter  
 Serre, Thomas  
 Shah, Mubarak  
 Shakhnarovich, Gregory  
 Shan, Ying  
 Shashua, Amnon  
 Shechtman, Eli  
 Sheikh, Yaser  
 Shekhovtsov, Alexander  
 Shet, Vinay  
 Shi, Jianbo  
 Shimshoni, Ilan  
 Shokoufandeh, Ali  
 Sigal, Leonid  
 Simon, Loic  
 Singara,ju, Dheeraaj  
 Singh, Maneesh  
 Singh, Vikas  
 Sinha, Sudipta  
 Sivic, Josef  
 Slabaugh, Greg  
 Smeulders, Arnold  
 Sminchisescu, Cristian  
 Smith, Kevin  
 Smith, William  
 Snavelly, Noah  
 Snoek, Cees  
 Soatto, Stefano  
 Sochen, Nir  
 Sochman, Jan  
 Sofka, Michal  
 Sorokin, Alexander  
 Southall, Ben  
 Souvenir, Richard  
 Srivastava, Anuj  
 Stauffer, Chris  
 Stein, Gideon  
 Strecha, Christoph  
 Sugimoto, Akihiro  
 Sullivan, Josephine  
 Sun, Deqing  
 Sun, Jian  
 Sun, Min  
 Sunkavalli, Kalyan  
 Suter, David  
 Svoboda, Tomas  
 Syeda-Mahmood,  
     Tanveer  
 Süsstrunk, Sabine  
 Tai, Yu-Wing  
 Takamatsu, Jun  
 Talbot, Hugues  
 Tan, Ping  
 Tan, Robby  
 Tanaka, Masayuki  
 Tao, Dacheng  
 Tappen, Marshall  
 Taylor, Camillo  
 Theobalt, Christian  
 Thonnat, Monique  
 Tieu, Kinh  
 Tistarelli, Massimo  
 Todorovic, Sinisa  
 Toreyin, Behcet Ugur  
 Torresani, Lorenzo  
 Torsello, Andrea  
 Toshev, Alexander  
 Trucco, Emanuele  
 Tschumperle, David  
 Tsin, Yanghai  
 Tu, Peter  
 Tung, Tony  
 Turek, Matt  
 Turk, Matthew  
 Tuzel, Oncel  
 Tyagi, Ambrish  
 Urschler, Martin  
 Urtasun, Raquel  
 Van de Weijer, Joost  
 van Gemert, Jan  
 van den Hengel, Anton  
 Vasilescu, M. Alex O.  
 Vedaldi, Andrea  
 Veeraraghavan, Ashok  
 Veksler, Olga  
 Verbeek, Jakob  
 Vese, Luminita  
 Vitaladevuni, Shiv  
 Vogiatzis, George  
 Vogler, Christian  
 Wachinger, Christian  
 Wada, Toshikazu  
 Wagner, Daniel  
 Wang, Chaohui  
 Wang, Hanzi  
 Wang, Hongcheng  
 Wang, Jue  
 Wang, Kai  
 Wang, Song  
 Wang, Xiaogang  
 Wang, Yang  
 Weese, Juergen  
 Wei, Yichen  
 Wein, Wolfgang  
 Welinder, Peter  
 Werner, Tomas  
 Westin, Carl-Fredrik

Wilburn, Bennett  
Wildes, Richard  
Williams, Oliver  
Wills, Josh  
Wilson, Kevin  
Wojek, Christian  
Wolf, Lior  
Wright, John  
Wu, Tai-Pang  
Wu, Ying  
Xiao, Jiangjian  
Xiao, Jianxiong  
Xiao, Jing  
Yagi, Yasushi  
Yan, Shuicheng  
Yang, Fei  
Yang, Jie  
Yang, Ming-Hsuan

Yang, Peng  
Yang, Qingxiong  
Yang, Ruigang  
Ye, Jieping  
Yeung, Dit-Yan  
Yezzi, Anthony  
Yilmaz, Alper  
Yin, Lijun  
Yoon, Kuk Jin  
Yu, Jingyi  
Yu, Kai  
Yu, Qian  
Yu, Stella  
Yuille, Alan  
Zach, Christopher  
Zaid, Harchaoui  
Zelnik-Manor, Lihi  
Zeng, Gang

Zhang, Cha  
Zhang, Li  
Zhang, Sheng  
Zhang, Weiwei  
Zhang, Wenchao  
Zhao, Wenyi  
Zheng, Yuanjie  
Zhou, Jinghao  
Zhou, Kevin  
Zhu, Leo  
Zhu, Song-Chun  
Zhu, Ying  
Zickler, Todd  
Zikic, Darko  
Zisserman, Andrew  
Zitnick, Larry  
Zivny, Stanislav  
Zuffi, Silvia

## Sponsoring Institutions

### Platinum Sponsor

INSTITUT NATIONAL  
DE RECHERCHE  
EN INFORMATIQUE  
ET EN AUTOMATIQUE



### Gold Sponsors



### Silver Sponsors





# Table of Contents – Part I

## Computational Imaging

Guided Image Filtering . . . . .	1
<i>Kaiming He, Jian Sun, and Xiaoou Tang</i>	
Analysis of Motion Blur with a Flutter Shutter Camera for Non-linear Motion . . . . .	15
<i>Yuanyuan Ding, Scott McCloskey, and Jingyi Yu</i>	
Error-Tolerant Image Compositing . . . . .	31
<i>Michael W. Tao, Micah K. Johnson, and Sylvain Paris</i>	
Blind Reflectometry . . . . .	45
<i>Fabiano Romeiro and Todd Zickler</i>	
Photometric Stereo for Dynamic Surface Orientations . . . . .	59
<i>Hyeonwoo Kim, Bennett Wilburn, and Moshe Ben-Ezra</i>	
Fully Isotropic Fast Marching Methods on Cartesian Grids . . . . .	73
<i>Vikram Appia and Anthony Yezzi</i>	

## Spotlights and Posters M1

Descattering Transmission via Angular Filtering . . . . .	86
<i>Jaewon Kim, Douglas Lanman, Yasuhiro Mukaigawa, and Ramesh Raskar</i>	
Flexible Voxels for Motion-Aware Videography . . . . .	100
<i>Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa G. Narasimhan</i>	
Learning PDEs for Image Restoration via Optimal Control . . . . .	115
<i>Risheng Liu, Zhouchen Lin, Wei Zhang, and Zhixun Su</i>	
Compressive Acquisition of Dynamic Scenes . . . . .	129
<i>Aswin C. Sankaranarayanan, Pavan K. Turaga, Richard G. Baraniuk, and Rama Chellappa</i>	
Scene Carving: Scene Consistent Image Retargeting . . . . .	143
<i>Alex Mansfield, Peter Gehler, Luc Van Gool, and Carsten Rother</i>	
Two-Phase Kernel Estimation for Robust Motion Deblurring . . . . .	157
<i>Li Xu and Jiaya Jia</i>	

Single Image Deblurring Using Motion Density Functions . . . . .	171
<i>Ankit Gupta, Neel Joshi, C. Lawrence Zitnick, Michael Cohen, and Brian Curless</i>	
An Iterative Method with General Convex Fidelity Term for Image Restoration . . . . .	185
<i>Miyoun Jung, Elena Resmerita, and Luminita Vese</i>	
One-Shot Optimal Exposure Control . . . . .	200
<i>David Ilstrup and Roberto Manduchi</i>	
Analyzing Depth from Coded Aperture Sets . . . . .	214
<i>Anat Levin</i>	
We Are Family: Joint Pose Estimation of Multiple Persons . . . . .	228
<i>Marcin Eichner and Vittorio Ferrari</i>	
Joint People, Event, and Location Recognition in Personal Photo Collections Using Cross-Domain Context . . . . .	243
<i>Dahua Lin, Ashish Kapoor, Gang Hua, and Simon Baker</i>	
Chrono-Gait Image: A Novel Temporal Template for Gait Recognition . . . . .	257
<i>Chen Wang, Junping Zhang, Jian Pu, Xiaoru Yuan, and Liang Wang</i>	
Robust Face Recognition Using Probabilistic Facial Trait Code . . . . .	271
<i>Ping-Han Lee, Gee-Sern Hsu, Szu-Wei Wu, and Yi-Ping Hung</i>	
A 2D Human Body Model Dressed in Eigen Clothing . . . . .	285
<i>Peng Guan, Oren Freifeld, and Michael J. Black</i>	
Self-Adapting Feature Layers . . . . .	299
<i>Pia Breuer and Volker Blanz</i>	
Face Recognition with Patterns of Oriented Edge Magnitudes . . . . .	313
<i>Ngoc-Son Vu and Alice Caplier</i>	
Spatial-Temporal Granularity-Tunable Gradients Partition (STGGP) Descriptors for Human Detection . . . . .	327
<i>Yazhou Liu, Shiguang Shan, Xilin Chen, Janne Heikkila, Wen Gao, and Matti Pietikainen</i>	
Being John Malkovich . . . . .	341
<i>Ira Kemelmacher-Shlizerman, Aditya Sankar, Eli Shechtman, and Steven M. Seitz</i>	
Facial Contour Labeling via Congealing . . . . .	354
<i>Xiaoming Liu, Yan Tong, Frederick W. Wheeler, and Peter H. Tu</i>	

Cascaded Confidence Filtering for Improved Tracking-by-Detection . . . . .	369
<i>Severin Stalder, Helmut Grabner, and Luc Van Gool</i>	
Inter-camera Association of Multi-target Tracks by On-Line Learned Appearance Affinity Models . . . . .	383
<i>Cheng-Hao Kuo, Chang Huang, and Ram Nevatia</i>	
Multi-person Tracking with Sparse Detection and Continuous Segmentation . . . . .	397
<i>Dennis Mitzel, Esther Horbert, Andreas Ess, and Bastian Leibe</i>	
Closed-Loop Adaptation for Robust Tracking . . . . .	411
<i>Jialue Fan, Xiaohui Shen, and Ying Wu</i>	
Gaussian-Like Spatial Priors for Articulated Tracking . . . . .	425
<i>Søren Hauberg, Stefan Sommer, and Kim Steenstrup Pedersen</i>	
Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow . . . . .	438
<i>Narayanan Sundaram, Thomas Brox, and Kurt Keutzer</i>	
Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings . . . . .	452
<i>Stefano Pellegrini, Andreas Ess, and Luc Van Gool</i>	
Globally Optimal Multi-target Tracking on a Hexagonal Lattice . . . . .	466
<i>Anton Andriyenko and Konrad Schindler</i>	
Discriminative Spatial Attention for Robust Tracking . . . . .	480
<i>Jialue Fan, Ying Wu, and Shengyang Dai</i>	
Object, Scene and Actions: Combining Multiple Features for Human Action Recognition . . . . .	494
<i>Nazli Ikizler-Cinbis and Stan Sclaroff</i>	
Representing Pairwise Spatial and Temporal Relations for Action Recognition . . . . .	508
<i>Pyyry Matikainen, Martial Hebert, and Rahul Sukthankar</i>	
Compact Video Description for Copy Detection with Precise Temporal Alignment . . . . .	522
<i>Matthijs Douze, Hervé Jégou, Cordelia Schmid, and Patrick Pérez</i>	
Modeling the Temporal Extent of Actions . . . . .	536
<i>Scott Satkin and Martial Hebert</i>	
Content-Based Retrieval of Functional Objects in Video Using Scene Context . . . . .	549
<i>Sangmin Oh, Anthony Hoogs, Matthew Turek, and Roderic Collins</i>	

Anomalous Behaviour Detection Using Spatiotemporal Oriented Energies, Subset Inclusion Histogram Comparison and Event-Driven Processing . . . . .	563
<i>Andrei Zaharescu and Richard Wildes</i>	
Tracklet Descriptors for Action Modeling and Video Analysis . . . . .	577
<i>Michalis Raptis and Stefano Soatto</i>	
Word Spotting in the Wild . . . . .	591
<i>Kai Wang and Serge Belongie</i>	
A Stochastic Graph Evolution Framework for Robust Multi-target Tracking . . . . .	605
<i>Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K. Roy-Chowdhury</i>	

## Spotlights and Posters M2

Backprojection Revisited: Scalable Multi-view Object Detection and Similarity Metrics for Detections . . . . .	620
<i>Nima Razavi, Juergen Gall, and Luc Van Gool</i>	
Multiple Instance Metric Learning from Automatically Labeled Bags of Faces . . . . .	634
<i>Mathieu Guillaumin, Jakob Verbeek, and Cordelia Schmid</i>	
Partition Min-Hash for Partial Duplicate Image Discovery . . . . .	648
<i>David C. Lee, Qifa Ke, and Michael Isard</i>	
Automatic Attribute Discovery and Characterization from Noisy Web Data . . . . .	663
<i>Tamara L. Berg, Alexander C. Berg, and Jonathan Shih</i>	
Learning to Recognize Objects from Unseen Modalities . . . . .	677
<i>C. Mario Christoudias, Raquel Urtasun, Mathieu Salzmann, and Trevor Darrell</i>	
Building Compact Local Pairwise Codebook with Joint Feature Space Clustering . . . . .	692
<i>Nobuyuki Morioka and Shin'ichi Satoh</i>	
Image-to-Class Distance Metric Learning for Image Classification . . . . .	706
<i>Zhengxiang Wang, Yiqun Hu, and Liang-Tien Chia</i>	
Extracting Structures in Image Collections for Object Recognition . . . . .	720
<i>Sandra Ebert, Diane Larlus, and Bernt Schiele</i>	
Size Does Matter: Improving Object Recognition and 3D Reconstruction with Cross-Media Analysis of Image Clusters . . . . .	734
<i>Stephan Gammeter, Till Quack, David Tingdahl, and Luc Van Gool</i>	

Avoiding Confusing Features in Place Recognition . . . . .	748
<i>Jan Knopp, Josef Sivic, and Tomas Pajdla</i>	
Semantic Label Sharing for Learning with Many Categories . . . . .	762
<i>Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba</i>	
Efficient Object Category Recognition Using Classemes . . . . .	776
<i>Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon</i>	
Practical Autocalibration . . . . .	790
<i>Riccardo Gherardi and Andrea Fusiello</i>	
<b>Author Index</b> . . . . .	803

# Guided Image Filtering

Kaiming He<sup>1</sup>, Jian Sun<sup>2</sup>, and Xiaoou Tang<sup>1,3</sup>

<sup>1</sup> Department of Information Engineering, The Chinese University of Hong Kong

<sup>2</sup> Microsoft Research Asia

<sup>3</sup> Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

**Abstract.** In this paper, we propose a novel type of explicit image filter - *guided filter*. Derived from a local linear model, the guided filter generates the filtering output by considering the content of a guidance image, which can be the input image itself or another different image. The guided filter can perform as an edge-preserving smoothing operator like the popular bilateral filter [1], but has better behavior near the edges. It also has a theoretical connection with the matting Laplacian matrix [2], so is a more generic concept than a smoothing operator and can better utilize the structures in the guidance image. Moreover, the guided filter has a fast and non-approximate linear-time algorithm, whose computational complexity is independent of the filtering kernel size. We demonstrate that the guided filter is both effective and efficient in a great variety of computer vision and computer graphics applications including noise reduction, detail smoothing/enhancement, HDR compression, image matting/feathering, haze removal, and joint upsampling.

## 1 Introduction

Most applications in computer vision and computer graphics involve the concept of image filtering to reduce noise and/or extract useful image structures. Simple explicit linear translation-invariant (LTI) filters like Gaussian filter, Laplacian filter, and Sobel filter are widely used in image blurring/sharpening, edge detection, and feature extraction [3]. LTI filtering also includes the process of solving a Poisson Equation, such as in high dynamic range (HDR) compression [4], image stitching [5], and image matting [6], where the filtering kernel is implicitly defined by the inverse of a homogenous Laplacian matrix.

The kernels of LTI filters are spatially invariant and independent of any image content. But in many cases, we may want to incorporate additional information from a given *guidance* image during the filtering process. For example, in colorization [7] the output chrominance channels should have consistent edges with the given luminance channel; in image matting [2] the output alpha matte should capture the thin structures like hair in the image. One approach to achieve this purpose is to optimize a quadratic function that directly enforces some constraints on the unknown output by considering the guidance image. The solution is then obtained by solving a large sparse matrix encoded with the information of the guidance image. This inhomogeneous matrix implicitly defines a *translation-variant* filtering kernel. This approach is widely used in many

applications, like colorization [7], image matting [2], multi-scale decomposition [8], and haze removal [9]. While this optimization-based approach often yields the state-of-the-art quality, it comes with the price of long computational time.

The other approach is to explicitly build the filter kernels using the guidance image. The bilateral filter, proposed in [10], made popular in [1], and later generalized in [11], is perhaps the most popular one of such filters. Its output at a pixel is a weighted average of the nearby pixels, where the weights depend on the intensity/color similarities in the guidance image. The guidance image can be the filter input itself [1] or another image [11]. The bilateral filter can smooth small fluctuations and preserve edges. While this filter is effective in many situations, it may have unwanted gradient reversal artifacts [12,13,8] near edges (further explained in Section 3.4). Its fast implementation is also a challenging problem. Recent techniques [14,15,16,17] rely on quantization methods to accelerate but may sacrifice the accuracy.

In this paper we propose a new type of explicit image filter, called *guided filter*. The filtering output is locally a linear transform of the guidance image. This filter has the edge-preserving smoothing property like the bilateral filter, but does not suffer from the gradient reversal artifacts. It is also related to the matting Laplacian matrix [2], so is a more generic concept and is applicable in other applications beyond the scope of "smoothing". Moreover, the guided filter has an  $O(N)$  time (in the number of pixels  $N$ ) *exact* algorithm for both gray-scale and color images. Experiments show that the guided filter performs very well in terms of both quality and efficiency in a great variety of applications, such as noise reduction, detail smoothing/enhancement, HDR compression, image matting/feathering, haze removal, and joint upsampling.

## 2 Related Work

### 2.1 Bilateral Filter

The bilateral filter computes the filter output at a pixel as a weighted average of neighboring pixels. It smooths the image while preserving edges. Due to this nice property, it has been widely used in noise reduction [18], HDR compression [12], multi-scale detail decomposition [19], and image abstraction [20]. It is generalized to the joint bilateral filter in [11], in which the weights are computed from another guidance image rather than the filter input. The joint bilateral filter is particularly favored when the filter input is not reliable to provide edge information, e.g., when it is very noisy or is an intermediate result. The joint bilateral filter is applicable in flash/no-flash denoising [11], image upsampling [21], and image deconvolution [22].

However, it has been noticed [12,13,8] that the bilateral filter may have the gradient reversal artifacts in detail decomposition and HDR compression. The reason is that when a pixel (often on an edge) has few similar pixels around it, the Gaussian weighted average is unstable. Another issue concerning the bilateral filter is its efficiency. The brute-force implementation is in  $O(Nr^2)$  time, which is prohibitively high when the kernel radius  $r$  is large. In [14] an

approximated solution is obtained in a discretized space-color grid. Recently,  $O(N)$  time algorithms [15,16] have been developed based on histograms. Adams et al. [17] propose a fast algorithm for color images. All the above methods require a high quantization degree to achieve satisfactory speed, but at the expense of quality degradation.

## 2.2 Optimization-Based Image Filtering

A series of approaches optimize a quadratic cost function and solve a linear system, which is equivalent to implicitly filtering an image by an inverse matrix. In image segmentation [23] and colorization [7], the affinities of this matrix are Gaussian functions of the color similarities. In image matting, a matting Laplacian matrix [2] is designed to enforce the alpha matte as a local linear transform of the image colors. This matrix is also applicable in haze removal [9]. The weighted least squares (WLS) filter in [8] adjusts the matrix affinities according to the image gradients and produces a halo-free decomposition of the input image. Although these optimization-based approaches often generate high quality results, solving the corresponding linear system is time-consuming.

It has been found that the optimization-based filters are closely related to the explicit filters. In [24] Elad shows that the bilateral filter is one Jacobi iteration in solving the Gaussian affinity matrix. In [25] Fattal defines the edge-avoiding wavelets to approximate the WLS filter. These explicit filters are often simpler and faster than the optimization-based filters.

## 3 Guided Filter

We first define a general linear translation-variant filtering process, which involves a guidance image  $I$ , an input image  $p$ , and an output image  $q$ . Both  $I$  and  $p$  are given beforehand according to the application, and they can be identical. The filtering output at a pixel  $i$  is expressed as a weighted average:

$$q_i = \sum_j W_{ij}(I)p_j, \quad (1)$$

where  $i$  and  $j$  are pixel indexes. The filter kernel  $W_{ij}$  is a function of the guidance image  $I$  and independent of  $p$ . This filter is linear with respect to  $p$ .

A concrete example of such a filter is the joint bilateral filter [11]. The bilateral filtering kernel  $W^{\text{bf}}$  is given by:

$$W_{ij}^{\text{bf}}(I) = \frac{1}{K_i} \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\sigma_s^2}\right) \exp\left(-\frac{|I_i - I_j|^2}{\sigma_r^2}\right). \quad (2)$$

where  $\mathbf{x}$  is the pixel coordinate, and  $K_i$  is a normalizing parameter to ensure that  $\sum_j W_{ij}^{\text{bf}} = 1$ . The parameters  $\sigma_s$  and  $\sigma_r$  adjust the spatial similarity and the range (intensity/color) similarity respectively. The joint bilateral filter degrades to the original bilateral filter [1] when  $I$  and  $p$  are identical.



### 3.1 Definition

Now we define the guided filter and its kernel. The key assumption of the guided filter is a local linear model between the guidance  $I$  and the filter output  $q$ . We assume that  $q$  is a linear transform of  $I$  in a window  $\omega_k$  centered at the pixel  $k$ :

$$q_i = a_k I_i + b_k, \forall i \in \omega_k, \quad (3)$$

where  $(a_k, b_k)$  are some linear coefficients assumed to be constant in  $\omega_k$ . We use a square window of a radius  $r$ . This local linear model ensures that  $q$  has an edge only if  $I$  has an edge, because  $\nabla q = a \nabla I$ . This model has been proven useful in image matting [2], image super-resolution [26], and haze removal [9].

To determine the linear coefficients, we seek a solution to (3) that minimizes the difference between  $q$  and the filter input  $p$ . Specifically, we minimize the following cost function in the window:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2). \quad (4)$$

Here  $\epsilon$  is a regularization parameter preventing  $a_k$  from being too large. We will investigate its significance in Section 3.2. The solution to (4) can be given by linear regression [27]:

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \quad (5)$$

$$b_k = \bar{p}_k - a_k \mu_k. \quad (6)$$

Here,  $\mu_k$  and  $\sigma_k^2$  are the mean and variance of  $I$  in  $\omega_k$ ,  $|\omega|$  is the number of pixels in  $\omega_k$ , and  $\bar{p}_k = \frac{1}{|\omega|} \sum_{i \in \omega_k} p_i$  is the mean of  $p$  in  $\omega_k$ .

Next we apply the linear model to all local windows in the entire image. However, a pixel  $i$  is involved in all the windows  $\omega_k$  that contain  $i$ , so the value of  $q_i$  in (3) is not the same when it is computed in different windows. A simple strategy is to average all the possible values of  $q_i$ . So after computing  $(a_k, b_k)$  for all patches  $\omega_k$  in the image, we compute the filter output by:

$$q_i = \frac{1}{|\omega|} \sum_{k: i \in \omega_k} (a_k I_i + b_k) \quad (7)$$

$$= \bar{a}_i I_i + \bar{b}_i \quad (8)$$

where  $\bar{a}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} a_k$  and  $\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$ .

With this modification  $\nabla q$  is no longer scaling of  $\nabla I$ , because the linear coefficients  $(\bar{a}_i, \bar{b}_i)$  vary spatially. But since  $(\bar{a}_i, \bar{b}_i)$  are the output of an average filter, their gradients should be much smaller than that of  $I$  near strong edges. In this situation we can still have  $\nabla q \approx \bar{a} \nabla I$ , meaning that abrupt intensity changes in  $I$  can be mostly maintained in  $q$ .

We point out that the relationship among  $I$ ,  $p$ , and  $q$  given by (5), (6), and (8) are indeed in the form of image filtering (1). In fact,  $a_k$  in (5) can be rewritten

as a weighted sum of  $p$ :  $a_k = \sum_j A_{kj}(I)p_j$ , where  $A_{ij}$  are the weights only dependent on  $I$ . For the same reason, we also have  $b_k = \sum_j B_{kj}(I)p_j$  from (6) and  $q_i = \sum_j W_{ij}(I)p_j$  from (8). It can be proven (see the supplementary materials) that the kernel weights can be explicitly expressed by:

$$W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k:(i,j) \in \omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}\right). \quad (9)$$

Some further computations show that  $\sum_j W_{ij}(I) = 1$ . No extra effort is needed to normalize the weights.

### 3.2 Edge-preserving Filtering

Fig. 1 (top) shows an example of the guided filter with various sets of parameters. We can see that it has the edge-preserving smoothing property. This can be explained intuitively as following. Consider the case that  $I = p$ . It is clear that if  $\epsilon = 0$ , then the solution to (4) is  $a_k = 1$  and  $b_k = 0$ . If  $\epsilon > 0$ , we can consider two cases:

Case 1: "Flat patch". If the image  $I$  is constant in  $\omega_k$ , then (4) is solved by  $a_k = 0$  and  $b_k = \bar{p}_k$ ;

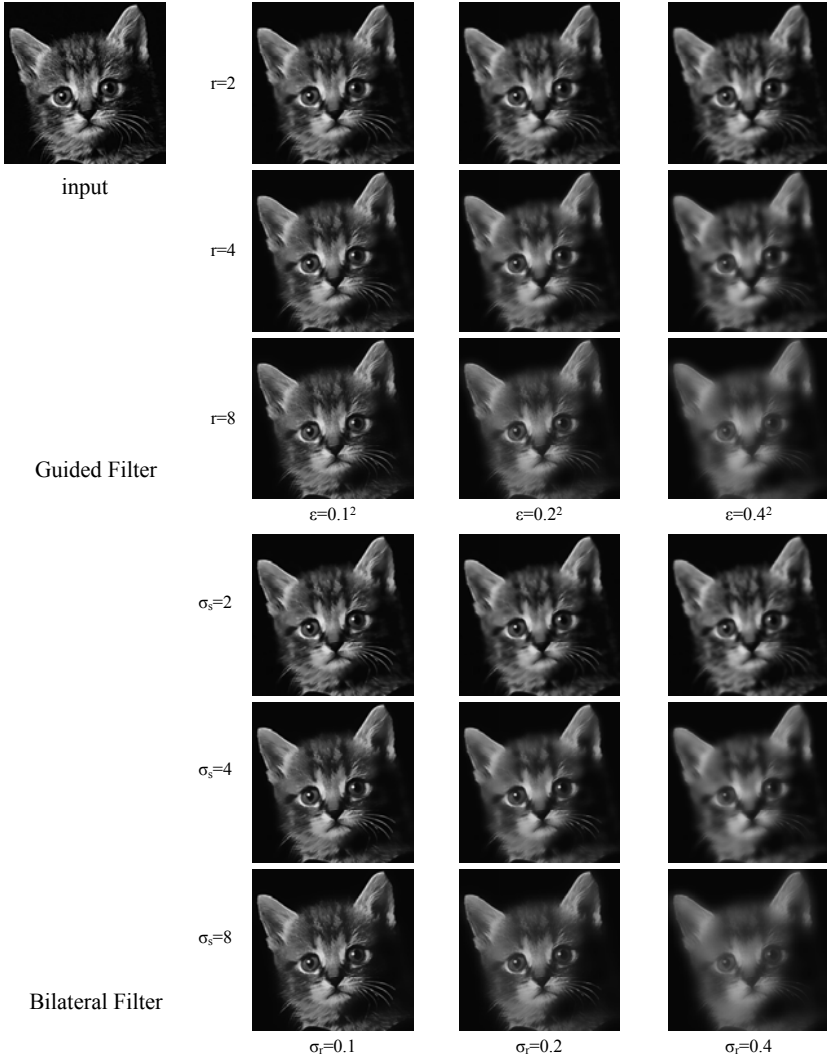
Case 2: "High variance". If the image  $I$  changes a lot within  $\omega_k$ , then  $a_k$  becomes close to 1 while  $b_k$  is close to 0.

When  $a_k$  and  $b_k$  are averaged to get  $\bar{a}_i$  and  $\bar{b}_i$ , combined in (8) to get the output, we have that if a pixel is in the middle of a "high variance" area, then its value is unchanged, whereas if it is in the middle of a "flat patch" area, its value becomes the average of the pixels nearby.

More specifically, the criterion of a "flat patch" or a "high variance" is given by the parameter  $\epsilon$ . The patches with variance ( $\sigma^2$ ) much smaller than  $\epsilon$  are smoothed, whereas those with variance much larger than  $\epsilon$  are preserved. The effect of  $\epsilon$  in the guided filter is similar with the range variance  $\sigma_r^2$  in the bilateral filter (2). Both parameters determine "what is an edge/a high variance patch that should be preserved". Fig. 1 (bottom) shows the bilateral filter results as a comparison.

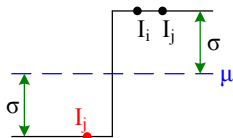
### 3.3 Filter Kernel

The edge-preserving smoothing property can also be understood by investigating the filter kernel (9). Take an ideal step edge of a 1-D signal as an example (Fig. 2). The terms  $I_i - \mu_k$  and  $I_j - \mu_k$  have the same sign (+/-) when  $I_i$  and  $I_j$  are on the same side of an edge, while they have opposite signs when the two pixels are on different sides. So in (9) the term  $1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}$  is much smaller (and close to zero) for two pixels on different sides than on the same sides. This means that the pixels across an edge are almost not averaged together. We can also understand the smoothing effect of  $\epsilon$  from (9). When  $\sigma_k^2 \ll \epsilon$  ("flat patch"), the kernel becomes  $W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k:(i,j) \in \omega_k} 1$ : this is a low-pass filter that biases neither side of an edge.

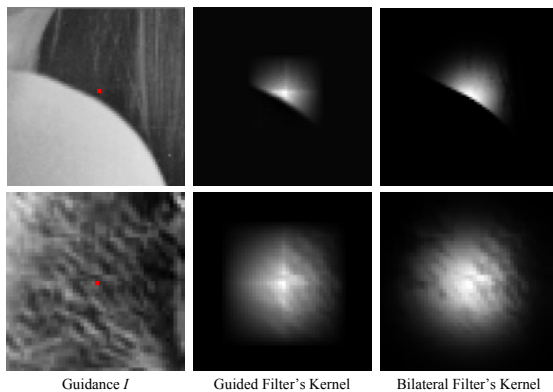


**Fig. 1.** The filtered images of a gray-scale input. In this example the guidance  $I$  is identical to the input  $p$ . The input image has intensity in  $[0, 1]$ . The input image is from [1].

Fig. 3 shows two examples of the kernel shapes in real images. In the top row are the kernels near a step edge. Like the bilateral kernel, the guided filter’s kernel assigns nearly zero weights to the pixels on the opposite side of the edge. In the bottom row are the kernels in a patch with small scale textures. Both filters average almost all the nearby pixels together and appear as low-pass filters.



**Fig. 2.** A 1-D example of an ideal step edge. For a window that exactly center on the edge, the variables  $\mu$  and  $\sigma$  are as indicated.



**Fig. 3.** Filter kernels. Top: a step edge (guided filter:  $r = 7, \epsilon = 0.1^2$ , bilateral filter:  $\sigma_s = 7, \sigma_r = 0.1$ ). Bottom: a textured patch (guided filter:  $r = 8, \epsilon = 0.2^2$ , bilateral filter:  $\sigma_s = 8, \sigma_r = 0.2$ ). The kernels are centered at the pixels denote by the red dots.

### 3.4 Gradient Preserving Filtering

Though the guided filter is an edge-preserving smoothing filter like the bilateral filter, it avoids the gradient reversal artifacts that may appear in detail enhancement and HDR compression. Fig. 4 shows a 1-D example of detail enhancement. Given the input signal (black), its edge-preserving smoothed output is used as a *base layer* (red). The difference between the input signal and the base layer is the *detail layer* (blue). It is magnified to boost the details. The enhanced signal (green) is the combination of the boosted detail layer and the base layer. An elaborate description of this method can be found in [12].

For the bilateral filter (Fig. 4 left), the base layer is not consistent with input signal at the edge pixels. This is because few pixels around them have similar colors, and the Gaussian weighted average has little statistical data and becomes unreliable. So the detail layer has great fluctuations, and the recombined signal has reversed gradients as shown in the figure. On the other hand, the guided filter (Fig. 4 right) better preserves the gradient information in  $I$ , because the gradient of the base layer is  $\nabla q \approx \bar{a} \nabla I$  near the edge. The shape of the edge is well maintained in the recombined layer.

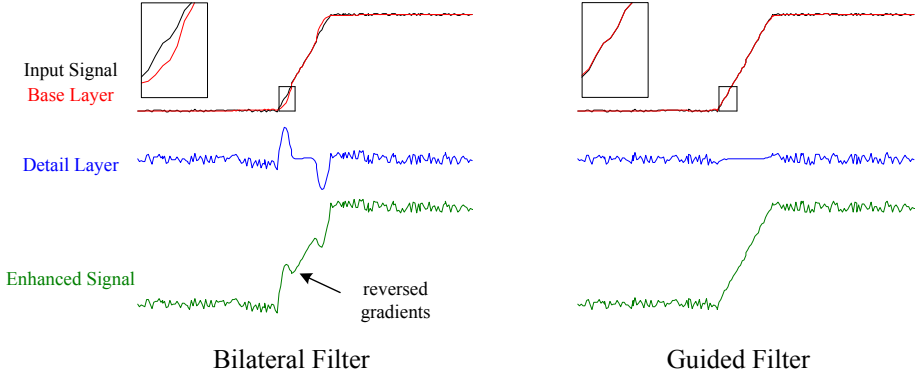


Fig. 4. 1-D illustration for detail enhancement. See the text for explanation

### 3.5 Relation to the Matting Laplacian Matrix

The guided filter can not only be used as a smoothing operator. It is also closely related to the matting Laplacian matrix [2]. This casts new insights into the guided filter and inspires some new applications.

In a closed-form solution to matting [2], the matting Laplacian matrix is derived from a local linear model. Unlike the guided filter which computes the local optimal for each window, the closed-form solution seeks a global optimal. To solve for the unknown alpha matte, this method minimizes the following cost function:

$$E(\alpha) = (\alpha - \beta)^T \Lambda (\alpha - \beta) + \alpha^T L \alpha, \quad (10)$$

where  $\alpha$  is the unknown alpha matte denoted in its matrix form,  $\beta$  is the constraint (e.g., a trimap),  $L$  is an  $N \times N$  matting Laplacian matrix, and  $\Lambda$  is a diagonal matrix encoded with the weights of the constraints. The solution to this optimization problem is given by solving a linear system:  $(L + \Lambda)\alpha = \Lambda\beta$ .

The elements of the matting Laplacian matrix are given by:

$$L_{ij} = \sum_{k:(i,j) \in \omega_k} \left( \delta_{ij} - \frac{1}{|\omega|} \left( 1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon} \right) \right). \quad (11)$$

where  $\delta_{ij}$  is the Kronecker delta. Comparing (11) with (9), we find that the elements of the matting Laplacian matrix can be directly given by the guided filter kernel weights:

$$L_{ij} = |\omega| (\delta_{ij} - W_{ij}), \quad (12)$$

Following the strategy in [24], we can further prove (see the supplementary materials) that the output of the guided filter is one Jacobi iteration in optimizing (10). If  $\beta$  is a reasonably good guess of the matte, we can run one Jacobi step and obtain an approximate solution to (10) by a guided filtering process:  $\alpha_i \approx \sum_j W_{ij}(I) \beta_j$ . In Section 4, we apply this property to image matting/feathering and haze removal.

### 3.6 O(N) Time Exact Algorithm

One more advantage of the guided filter over the bilateral filter is that it automatically has an O(N) time exact algorithm. O(N) time implies that the time complexity is independent of the window radius  $r$ , so we are free to use arbitrary kernel sizes in the applications.

The filtering process in (11) is a translation-variant convolution. Its computational complexity increases when the kernel becomes larger. Instead of directly performing the convolution, we compute the filter output from its definition (5) (6) (8). All the summations in these equations are box filters ( $\sum_{i \in \omega_k} f_i$ ). We apply the O(N) time Integral Image technique [28] to calculate the output of a box filter. So the guided filter can be computed in O(N) time.

The O(N) time algorithm can be easily extended to RGB color guidance images. Filtering using color guidance images is necessary when the edges or details are not discriminable in any single channel. To generalize to a color guidance image, we rewrite the local linear model (3) as:

$$q_i = \mathbf{a}_k^T \mathbf{I}_i + b_k, \forall i \in \omega_k. \quad (13)$$

Here  $\mathbf{I}_i$  is a  $3 \times 1$  color vector,  $\mathbf{a}_k$  is a  $3 \times 1$  coefficient vector,  $q_i$  and  $b_k$  are scalars. The guided filter for color guidance images becomes:

$$\mathbf{a}_k = (\Sigma_k + \epsilon \mathbf{U})^{-1} \left( \frac{1}{|\omega|} \sum_{i \in \omega_k} \mathbf{I}_i p_i - \mu_k \bar{p}_k \right) \quad (14)$$

$$b_k = \bar{p}_k - \mathbf{a}_k^T \mu_k \quad (15)$$

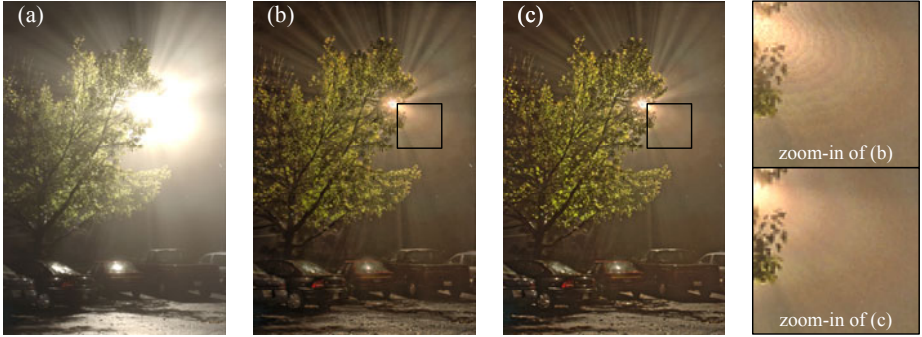
$$q_i = \mathbf{a}_i^T \mathbf{I}_i + \bar{b}_i. \quad (16)$$

Here  $\Sigma_k$  is the  $3 \times 3$  covariance matrix of  $\mathbf{I}$  in  $\omega_k$ , and  $\mathbf{U}$  is a  $3 \times 3$  identity matrix. The summations are still box filters and can be computed in O(N) time.

We experiment the running time in a laptop with a 2.0Hz Intel Core 2 Duo CPU. For the gray-scale guided filter, it takes 80ms to process a 1-megapixel image. As a comparison, the O(N) time bilateral filter in [15] requires 42ms using a histogram of 32 bins, and 85ms using 64 bins. Note that the guided filter algorithm is non-approximate and applicable for data of high bit-depth, while the O(N) time bilateral filter may have noticeable quantization artifacts (see Fig. 5). The algorithm in [16] requires 1.2 seconds per megapixel using 8 bins (using the public code on the authors' website). For RGB guidance images, the guided filter takes about 0.3s to process a 1-megapixel image. The algorithm for high-dimensional bilateral filter in [16] takes about 10 seconds on average to process per 1-megapixel RGB image.

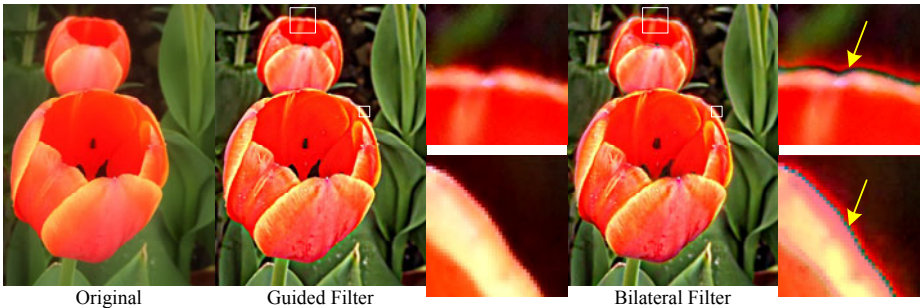
## 4 Applications and Experimental Results

In this section, we apply the guided filter to a great variety of computer vision and graphics applications.



**Fig. 5.** Quantization artifacts of  $O(N)$  time bilateral filter. (a) Input HDR image (32bit float, displayed by linear scaling). (b) Compressed image using the  $O(N)$  bilateral filter in [15] (64 bins). (c) Compressed image using the guided filter. This figure is best viewed in the electronic version of this paper.

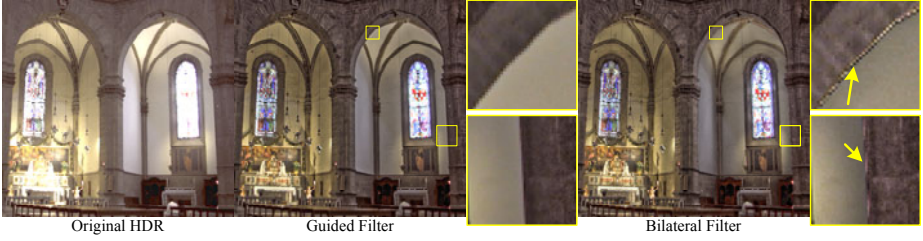
**Detail Enhancement and HDR Compression.** The method for detail enhancement is described in Section 3.4. For HDR compression, we compress the base layer instead of magnifying the detail layer. Fig. 6 shows an example for detail enhancement, and Fig. 7 shows an example for HDR Compression. The results using the bilateral filter are also provided. As shown in the zoom-in patches, the bilateral filter leads to gradient reversal artifacts.



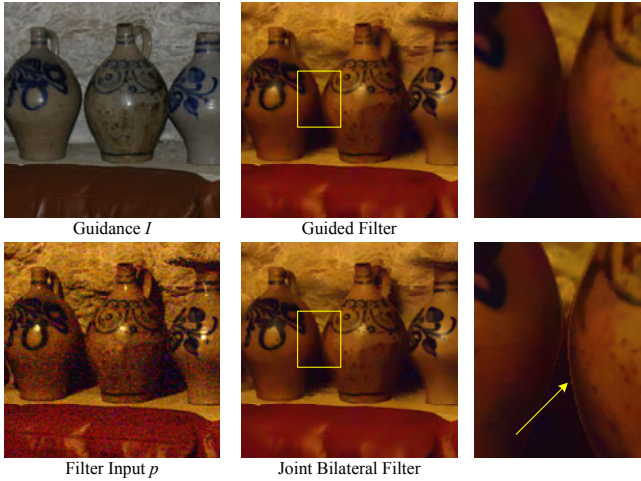
**Fig. 6.** Detail enhancement. The parameters are  $r = 16$ ,  $\epsilon = 0.1^2$  for the guided filter, and  $\sigma_s = 16$ ,  $\sigma_r = 0.1$  for the bilateral filter. The detail layer is boosted  $\times 5$ .

**Flash/No-flash Denoising.** In [11] it is proposed to denoise a no-flash image under the guidance of its flash version. Fig. 8 shows a comparison of using the joint bilateral filter and the guided filter. The gradient reversal artifacts are noticeable near some edges in the joint bilateral filter result.

**Matting/Guided Feathering.** We apply the guided filter as *guided feathering*: a binary mask is refined to appear an alpha matte near the object boundaries



**Fig. 7.** HDR compression. The parameters are  $r = 15$ ,  $\epsilon = 0.12^2$  for the guided filter, and  $\sigma_s = 15$ ,  $\sigma_r = 0.12$  for the bilateral filter.

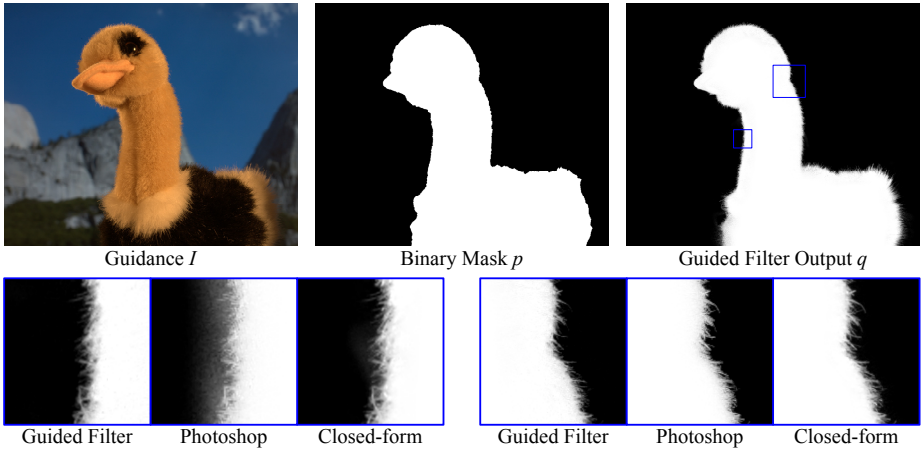


**Fig. 8.** Flash/no-flash denoising. The parameters are  $r = 8$ ,  $\epsilon = 0.2^2$  for the guided filter, and  $\sigma_s = 8$ ,  $\sigma_r = 0.2$  for the joint bilateral filter.

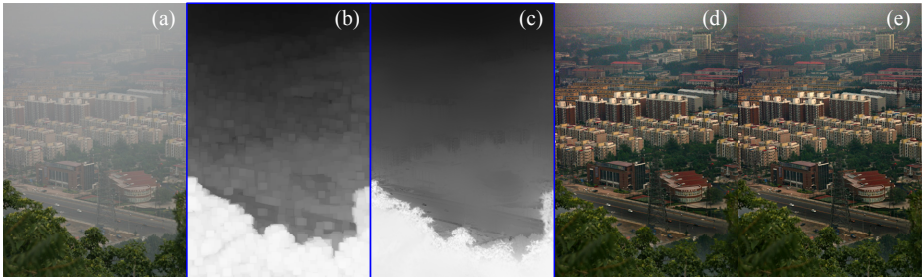
(Fig. 9). The binary mask can be obtained from graph-cut or other segmentation methods, and is used as the filter input  $p$ . The guidance  $I$  is the color image. A similar function “Refine Edge” can be found in the commercial software Adobe Photoshop CS4. We can also compute an accurate matte using the closed-form solution [2]. In Fig. 9 we compare our results with the Photoshop Refine Edge and the closed-form solution. Our result is visually comparable with the closed-form solution in this short hair case. Both our method and Photoshop provide fast feedback (<1s) for this 6-mega-pixel image, while the closed-form solution takes about two minutes to solve a huge linear system.

**Single Image Haze Removal.** In [9] a haze transmission map is roughly estimated using a dark channel prior, and is refined by solving the matting Laplacian matrix. On the contrary, we simply filter the raw transmission map under the





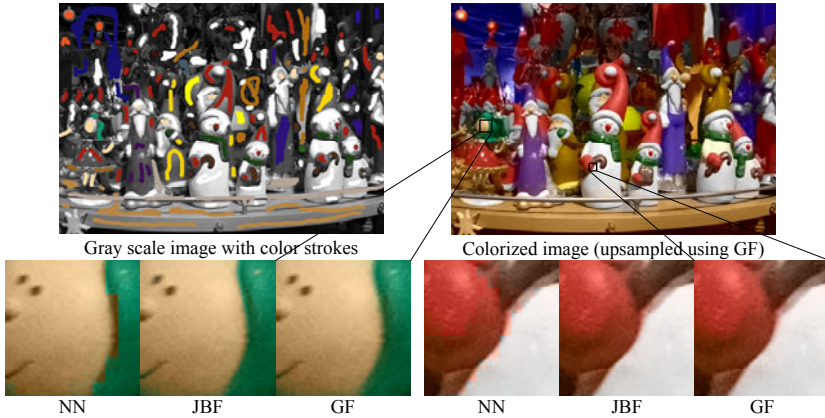
**Fig. 9.** Guided Feathering. A binary mask  $p$  is filtered under the guidance of  $I$ . In the zoom-in patches, we compare with the Photoshop Refine Edge function and the closed-form matting. For closed-form matting, we erode and dilate the mask to obtain a trimap. The parameters are  $r = 60$ ,  $\epsilon = 10^{-6}$  for the guided filter.



**Fig. 10.** Haze Removal. (a) Hazy image. (b) Raw transmission map [9]. (c) The raw transmission map is refined by the guided filter ( $r = 20$ ,  $\epsilon = 10^{-3}$ ). (e) Recovered image using (c). (d) The result in [9].

guidance of the hazy image. The results are visually similar (Fig. 10). The guided filter takes about 0.1s to process this  $600 \times 400$  color image, but the running time is over 10 seconds as reported in [9].

**Joint Upsampling.** Joint upsampling [21] is to upsample an image under the guidance of another image. Taking the application of colorization [7] as an example. A gray-scale luminance image is colorized through an optimization process. To reduce the running time, the chrominance channels are solved at a coarse resolution and upsampled under the guidance of the full resolution luminance image by the joint bilateral filter [21]. This upsampling process can also be performed by the guided filter. The result is visually comparable (Fig. 11).



**Fig. 11.** Joint Upsampling for Colorization. The upsampling methods includes: nearest-neighbor (NN), joint bilateral filter (JBF), and guided filter (GF).

## 5 Discussion and Conclusion

In this paper, we have presented a novel filter which is widely applicable in computer vision and graphics. Different from the recent trend towards accelerating the bilateral filter [14, 15, 16, 17], we define a new type of filter that shares the nice property of edge-preserving smoothing but can be computed efficiently and exactly. Our filter is more generic and can handle some applications beyond the concept of "smoothing". Since the local linear model (3) can be regarded as a simple case of learning, other advanced models/features might be applied to obtain new filters.

As a locally based operator, the guided filter is not directly applicable for sparse inputs like strokes. It also shares a common limitation of other explicit filter - it may have halos near some edges. In fact, it is ambiguous for a low-level and local operator to determine which edge should be smoothed and which should be preserved. Unsuitably smoothing an edge will result in halos near it. However, we believe that the simplicity and efficiency of the guided filter still make it beneficial in many situations.

## References

1. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV (1998)
2. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. In: CVPR (2006)
3. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice-Hall, Englewood Cliffs (2002)
4. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. In: SIGGRAPH (2002)
5. Pérez, P.: Poisson image editing. In: SIGGRAPH (2003)

6. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. In: SIGGRAPH (2004)
7. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: SIGGRAPH (2004)
8. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In: SIGGRAPH (2008)
9. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: CVPR (2009)
10. Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: Mustererkennung 1995, DAGM-Symposium, vol. 17, pp. 538–545. Springer, Heidelberg (1995)
11. Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., Toyama, K.: Digital photography with flash and no-flash image pairs. In: SIGGRAPH (2004)
12. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. In: SIGGRAPH (2002)
13. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. In: SIGGRAPH (2006)
14. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 568–580. Springer, Heidelberg (2006)
15. Porikli, F.: Constant time  $o(1)$  bilateral filtering. In: CVPR (2008)
16. Yang, Q., Tan, K.H., Ahuja, N.: Real-time  $o(1)$  bilateral filtering. In: CVPR (2009)
17. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian kd-trees for fast high-dimensional filtering. In: SIGGRAPH (2009)
18. Liu, C., Freeman, W.T., Szeliski, R., Kang, S.B.: Noise estimation from a single image. In: CVPR (2006)
19. Fattal, R., Agrawala, M., Rusinkiewicz, S.: Multiscale shape and detail enhancement from multi-light image collections. In: SIGGRAPH (2007)
20. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. In: SIGGRAPH (2006)
21. Kopf, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. In: SIGGRAPH (2007)
22. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. In: SIGGRAPH (2008)
23. Weiss, Y.: Segmentation using eigenvectors: A unifying view. In: ICCV (1999)
24. Elad, M.: On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing* (2002)
25. Fattal, R.: Edge-avoiding wavelets and their applications. In: SIGGRAPH (2009)
26. Zomet, A., Peleg, S.: Multi-sensor super resolution. In: *IEEE Workshop on Applications of Computer Vision* (2002)
27. Draper, N., Smith, H.: *Applied Regression Analysis*, 2nd edn. John Wiley, Chichester (1981)
28. Crow, F.: Summed-area tables for texture mapping. In: SIGGRAPH (1984)

# Analysis of Motion Blur with a Flutter Shutter Camera for Non-linear Motion

Yuanyuan Ding<sup>1</sup>, Scott McCloskey<sup>2</sup>, and Jingyi Yu<sup>1</sup>

<sup>1</sup> University of Delaware, Newark, DE, USA

<sup>2</sup> Honeywell Labs, Golden Valley, MN, USA

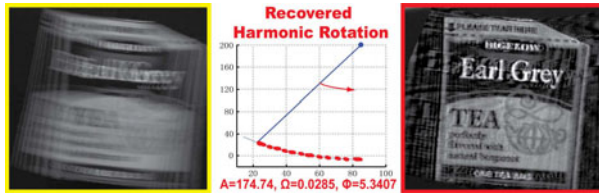
**Abstract.** Motion blurs confound many computer vision problems. The fluttered shutter (FS) camera [1] tackles the motion deblurring problem by emulating invertible broadband blur kernels. However, existing FS methods assume known constant velocity motions, e.g., via user specifications. In this paper, we extend the FS technique to general 1D motions and develop an automatic motion-from-blur framework by analyzing the image statistics under the FS.

We first introduce a fluttered-shutter point-spread-function (FS-PSF) to uniformly model the blur kernel under general motions. We show that many commonly used motions have closed-form FS-PSFs. To recover the FS-PSF from the blurred image, we present a new method by analyzing image power spectrum statistics. We show that the Modulation Transfer Function of the 1D FS-PSF is statistically correlated to the blurred image power spectrum along the motion direction. We then recover the FS-PSF by finding the motion parameters that maximize the correlation. We demonstrate our techniques on a variety of motions including constant velocity, constant acceleration, and harmonic rotation. Experimental results show that our method can automatically and accurately recover the motion from the blurs captured under the fluttered shutter.

## 1 Introduction

Restoring motion blurred images is a challenging task as it relies on both accurate kernel estimation and robust deconvolution. Most existing approaches assume the blurs are caused by constant velocity motion and model the kernel as a box filter. Tremendous efforts have been focused on designing robust deconvolution methods, from the earlier approaches based on regularization [2] to the latest ones using image statistics [3] and edge priors [4]. However, since the box filter destroys high-frequency features that are difficult to recover post-capture, results using these deconvolution methods may still contain strong artifacts.

Several computational photography methods have recently been proposed to change the frequency profile of the blur kernel. The fluttered shutter (FS) camera developed by Raskar et al. [1] opens and closes the shutter during the exposure process according to a pre-determined sequence. The pseudo-random sequence creates a broad-band filter that preserves high frequency details and is robust



**Fig. 1.** Motion estimation and deblurring of harmonic rotation using our approach.

to deconvolve. However, most existing fluttered shutter methods assume known constant velocity motions and rely on either user inputs [1] or alpha matting [5] to find the blur extent.

This paper addresses two fundamental problems when using the fluttered shutter: 1) how to apply the FS to handle a broader class of motions and 2) how to automatically recover the motion from the blurred image. For the first, we introduce a new fluttered-shutter point-spread-function (FS-PSF). FS-PSF uniformly models the blur kernel of arbitrary motions by computing how long each pixel gets exposed to the moving scene point throughout the shutter sequence. We show that many common motions such as constant velocity, acceleration, and harmonic rotation have closed-form FS-PSFs.

For the second, we present a new motion-from-blur method based on image power spectrum statistics. Schaaf and Hateren [6] have shown that circular power spectrum statistics of blur-free images follow the  $1/\omega$ -exponent model. We extend their analysis to model the linear power spectrum of motion blurred images captured under the FS. We show that the Modulation Transfer Function (MTF) of the 1D FS-PSF should be strongly correlated to the linear statistics of the blurred image along the motion direction. We then develop a matching algorithm using a sign-of-derivative metric to find the motion parameters that yield the strongest correlation. We demonstrate our techniques on real images of various motion types. We show that our method can automatically and accurately recover the motion parameters from blurs under the fluttered shutter. Furthermore, the recovered motion can be used to modify the initial shutter sequence with improved invertibility in cases that have not previously been addressed in literature on coded exposure. Our specific contributions are:

1. A new motion-from-blur framework analyzing Fourier image statistics.
2. A closed-form formulation of the fluttered shutter point-spread-function (FS-PSF) to model general 1D motion blurs under the FS.
3. A new image statistics analysis that correlates the MTF of the FS-PSF with linear power spectrum statistics of the blurred image.
4. A sign-of-derivative matching algorithm to find the motion parameters that maximize the correlation. Our method also leads to the new design of the motion-aware fluttered shutters.

## 2 Related Work

Existing algorithms related to motion blur have focused on three main aspects: blur kernel (PSF) estimation, image deconvolution and, most recently, image acquisition.

**PSF Estimation:** PSF estimation from a single image is known to be ill-posed. Existing methods make it tractable in a number of ways. Yuan et al. [7], for example, use a blurred/noisy image pair of the same scene. Other approaches employ regularization, such as the classical Wiener filter [2]. Still other approaches tackle the PSF estimation problem by constraining the space of potential PSFs. Assuming that blur arises from a traditional shutter with linear, constant-velocity motion constrains the potential PSFs to box filters. The cepstrum methods [8,9,10] have been proposed to characterize the motion by the number and position of zeros in the image power spectrum. However, these methods cannot be applied to fluttered shutter images that are acquired specifically to avoid such zeros. Recently, Dai and Wu [11] treat motion blurs as an alpha matte for estimating the PSF. Agrawal and Xu [5] apply a similar approach on the fluttered shutter. The implicit assumption in alpha-matte-based PSF estimation is the existence of high-contrast edges in the latent sharp image. Since the alpha matte only provides the blur extent, such methods cannot distinguish between the infinite number of velocity/acceleration combinations that might produce that extent.

**Image Deconvolution:** Numerous methods in the category of blind deconvolution [12] have been presented to mitigate the effects of motion or optical blur in images. Most motion deconvolution methods are based on the assumption that the object is moving along a straight line with constant velocity, in which case the PSF is a 1D box filter. Levin [4] examines the consequences of this type of blur on image statistics in order to perform blind deconvolution on blurred regions. It is also well understood that the magnitude of the Fourier transform of such a PSF has many zero points, where the frequency cannot be fully recovered. These missing frequencies lead to artifacts when using standard deconvolution. Though the scene's content at these spatial frequencies is irrecoverable, outside information in the form of gradient or edge priors [13,14,15,3,16] can be used to produce visually pleasing images.

**Acquisition:** Sharp image acquisition of fast-moving can also be achieved using short exposure duration with high-powered flashes, which is impractical in most settings. Many modern digital cameras have adaptive optical elements controlled by inertial sensors to reduce the effects of moderate camera motion due to hand shakes. Using video with varying exposure durations, Agrawal et al. [17] capture multiple images with partial coverage of the spatial frequency spectrum, which are combined to produce a single sharp image with coverage of all spatial frequencies. Hybrid cameras [18,19,20] use additional images/video to obviate or simplify the kernel estimation step.

Our work is motivated by the Flutter Shutter (FS) method by Raskar et al. [1], in which a single image is acquired by randomly opening and closing the

camera’s shutter during image capture. For constant velocity motion, the resulting blur kernel is invertible and standard image deconvolution can be directly used for deblurring. However, existing FS techniques assume known motion extent, e.g., via user specifications. In contrast, we set out to actively recover the motion from the blur. Our work is also related to Depth-from-Defocus (DfD) methods based on the coded apertures [21]. Although both DfD and motion estimation can be formulated as kernel estimation problems, motion blur kernels are usually complex yet spatially-invariant whereas defocus blur kernels are simple but spatially variant. As a result, motion estimation methods can uniformly treat groups of pixels, e.g., via image statistics [3] while DfD techniques rely on other types of priors such as smoothness or edges [22,23]. In this paper we analyze image statistics under the fluttered shutter for motion estimation.

### 3 Fluttered Shutter Point Spread Function (FS-PSF)

We start with defining the point-spread-function under the fluttered shutter that we call FS-PSF. We represent the shutter’s fluttering pattern as a sequence of chops with 1/0 values denoting the open/closed shutter states. We set every chop to have the same period  $w_{chop}$  and will use  $w_{chop}$  as the time unit  $t$  in the following analysis. Let  $S(t)$  denote the flutter sequence, we have:

$$S(t) = \begin{cases} 0 & \text{shutter closed} \\ 1 & \text{shutter open} \end{cases}, t = 1, 2, 3, \dots, M_s \quad (1)$$

where  $t$  represents time,  $M_s$  is the number of chops in the sequence, and  $E_s = w_{chop} \sum_{t=1}^{M_s} S(t)$  is the total exposure time.

The normalized FS-PSF  $p(x)$  describes how much each pixel  $x$  gets exposed to a moving scene point  $Q$ . Therefore, it is a function of both the shutter sequence  $S(t)$  and the motion of  $Q$ . To simplify our analysis, we adopt the same assumption in [1] that the moving object is frontal-planar and the FS-PSF is spatially-invariant. We measure the motion parameters such as displacement, velocity, and acceleration in unit of pixels, e.g., velocity as pixel/chop.

Recall that pixel  $x$  gets exposed to  $Q$  when  $Q$ ’s image passes through  $x$ . The exposure duration  $w(x)$  is inverse proportional to  $Q$ ’s velocity  $\nu(x)$  as:

$$w(x) = \frac{1}{\nu(x)} \quad (2)$$

Notice that, for general motions, it is natural to describe the velocity and displacement in terms of  $t$ . Thus, we can rewrite  $w(x) = \frac{1}{\nu(t(x))}$ , where  $t(x)$  is the inverse of the displacement function  $x(t)$ . In this paper, we assume that  $x(t)$  is monotonic throughout the shutter sequence, i.e., there is no back and forth motion, so that  $x(t)$  is invertible.

Finally, we combine the shutter sequence and the exposure  $w(x)$  to compute the un-normalized FS-PSF  $p_0(x)$  [1] as:

---

<sup>1</sup>  $p_0(x)$  is later be normalized to FS-PSF  $p(x)$ .

$$p_0(x) = S(t(x))w(t(x)) = \frac{S(t(x))}{\nu(t(x))} \quad (3)$$

Eq. (3) indicates that the FS-PSF can be viewed as an envelope of  $w(x)$  sampled by the shutter pattern  $S(t)$  as shown in Fig. 2. To derive the FS-PSF for arbitrary motions, we simply need to derive  $t(x)$ .

### 3.1 Constant Velocity

For constant velocity motion at  $\nu_c$  pixels/chop, we assume the first exposed pixel is the 0-th pixel, and we have  $x(t) = \nu_c \cdot t$  and  $t(x) = x/\nu_c$ . The FS-PSF is thus:

$$p_0(x) = \frac{S(t(x))}{\nu_c} = \frac{S(\frac{x}{\nu_c})}{\nu_c}, \quad x = 1, \dots, \nu_c M_s \quad (4)$$

Since the last exposed pixel coordinate has  $x(M_s) = \nu_c M_s$ , we can compute the normalized FS-PSF as:

$$p(x) = \frac{p_0(x)}{\sum_{x=1}^{\nu_c M_s} p_0(x)} = \frac{S(\frac{x}{\nu_c})}{\nu_c E_s}, \quad x = 1, \dots, \nu_c M_s \quad (5)$$

Eq. (5) indicates that varying the velocity  $\nu_c$  will result in spatial scaling in the PSF  $p(x)$ , while the envelope of  $p(x)$  remains as a rectangle. An example is shown in Fig. 2(left); the discretization of  $x$  in pixels may result in non-integer values. Our FS-PSF approximates non-integer pixels as the closest integer pixels with partial exposure intensity.

Recall that existing methods [1, 5] directly treat the shutter sequence as the PSF, i.e.,  $p(x) = S(x)$ . It is a special case of Eq. (4) where  $\nu_c = 1$  pixel/chop, and was achieved by manually re-sampling the captured image.

### 3.2 Constant Acceleration

Let  $\nu_s$  and  $\nu_e$  denote the velocity of  $Q$  at the start and end of the shutter sequence. The acceleration  $a$  can be computed as:  $a = \frac{\nu_e - \nu_s}{M_s}$ . The velocity  $\nu(t)$  and displacement  $x(t)$  are:

$$\nu(t) = \nu_s + a \cdot t, \quad x(t) = \nu_s \cdot t + \frac{a}{2} \cdot t^2 \quad (6)$$

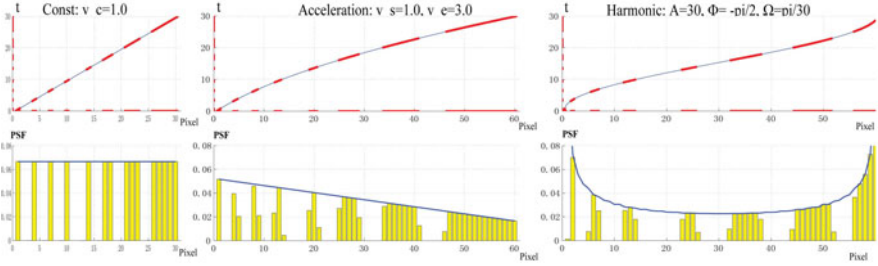
We can invert Eq. (6) to compute  $t(x)$  as:

$$t(x) = \frac{-\nu_s + \sqrt{\nu_s^2 + 2a \cdot x}}{a} \quad (7)$$

Finally, we can derive its FS-PSF using Eq. (3):

$$p_0(x) = \frac{S(t(x))}{\nu(t(x))} = \frac{S(\frac{-\nu_s + \sqrt{\nu_s^2 + 2a \cdot x}}{a})}{\sqrt{\nu_s^2 + 2a \cdot x}} \quad (8)$$





**Fig. 2.** FS-PSFs of common motions: constant velocity (left), constant acceleration (middle), and harmonic motion (right). Top row shows the time-velocity function sampled by the shutter (in red). Bottom row shows the corresponding FS-PSF.

The envelope of the constant acceleration FS-PSFs can be approximated as a trapezoid, as shown in Fig. 2 (middle); changing the starting velocity or the acceleration varies the slope and the shape of the trapezoid. We use  $v_s$  and  $a$  as the parameters for constant acceleration motion.

### 3.3 Linear Harmonic Motion

Linear harmonic motion is a periodic motion, where an object oscillates about an equilibrium position in a sinusoidal pattern, such as the commonly studied spring-mass system and the pendulum (recall Fig. 1).

We parameterize the linear harmonic motion by the amplitude  $A$ , the angular speed  $\Omega$ , and the initial phase  $\Phi$ . We first compute  $x$  and  $v$  as functions of  $t$ :

$$x(t) = A \sin(\Omega t + \Phi), \quad v(t) = A\Omega \cos(\Omega t + \Phi) \quad (9)$$

we solve  $t$  as an inverse function of  $x$  from Eq. (9):

$$t(x) = \frac{\arcsin(\frac{x}{A} - \Phi)}{\Omega} \quad (10)$$

Finally, we re-write Eq. (3) and compute the corresponding FS-PSF:

$$p_0(x) = \frac{S(t(x))}{v(t(x))} = \frac{S(\frac{\arcsin(\frac{x}{A} - \Phi)}{\Omega})}{A\Omega \cos(\Omega \cdot \frac{\arcsin(\frac{x}{A} - \Phi)}{\Omega} + \Phi)} \quad (11)$$

Fig. 2(right) illustrates harmonic motion with  $A = 30$ ,  $\Phi = \pi/2$ ,  $\Omega = \pi/30$ .

## 4 Recovering Motion PSFs

We have shown many commonly observed motions have closed-form PSFs. Our goal is to recover the FS-PSF by analyzing blurred images. Recall that the process of motion blur can be modeled as standard convolution:

$$i(x, y) = j \otimes p(x, y) + n(x, y) \quad (12)$$

where  $\otimes$  is the convolution operator,  $j$  is the latent sharp image,  $i$  is the degraded image,  $p$  is the blur kernel, and  $n$  is noise.

If we ignore  $n$ , we can model the amplitude spectrum of Eq. (12) as:

$$|I| = |JP| = |J||P| \quad (13)$$

where  $i/I$ ,  $j/J$ , and  $p/P$  are Fourier pairs, and  $|\cdot|$  is the modulus operator.  $|P|$  is also called the Modulation Transfer Function for 1D PSFs.

#### 4.1 Power Spectrum Statistics

Our FS-PSF estimation algorithm is based on power spectrum statistics in natural images. van der Schaaf and van Hateren [6] have shown that, for a natural image  $j$  without motion blur, its circular power spectrum statistics follows the  $1/\omega$ -exponent model: if we parameterize  $|J|$  in polar coordinates  $(\omega, \phi)$  where  $\omega$  is the radius (absolute frequency) and  $\phi$  is the angle, we can average  $|J|$  over  $\phi$  for every  $\omega$  and the resulting circular averaged power spectrum  $\text{circ}_\omega(|J|) \approx \frac{C}{\omega^m}$ , where  $m$  and  $C$  are constants. Statistically, if we assume every frequency is an independent and identically distributed random variable, circular statistics reveals that the expected value of  $|J(u, v)|$  is:

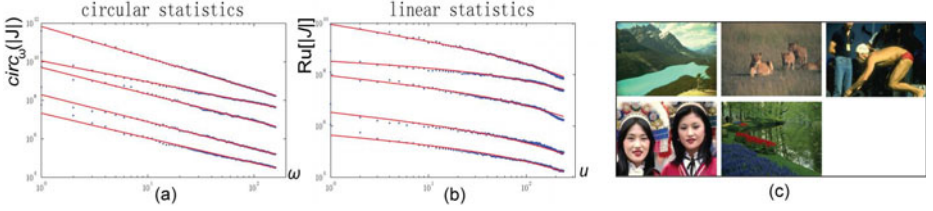
$$E[|J(u, v)|] = \frac{C}{(u^2 + v^2)^{m/2}} \quad (14)$$

Fig. 3(a) shows example traces of the power spectra of five natural images.

Our goal is to use power spectrum statistics to recover the FS-PSF from the blurred image  $i$ . In this paper, we assume the motion type (constant velocity, acceleration, etc.) is known and we focus on recovering its corresponding motion parameters  $\alpha$ . Given an candidate  $\alpha$ , we can compute the closed-form FS-PSF  $p$  as shown in Sec. 3 and calculate its MTF  $|P|$ . The latent image power spectrum  $|J|$  can then be computed as  $|I|/|P|$  from Eq. (13). If  $\alpha$  is the correct motion estimate,  $J$  should be motion blur free and its circular averaged power spectrum  $\text{circ}_\omega(|J|)$  should follow  $\frac{C}{\omega^m}$  distribution. A naive approach for testing if  $\alpha$  is a good motion estimation, then, would be to check if  $\text{circ}_\omega(|I|/|P|)$  has a  $\frac{C}{\omega^m}$  distribution. However, since  $\text{circ}_\omega(|J|)$  only represents the statistics of  $|J|$ , incorrectly estimated motion parameters may still produce such distributions. Therefore, we set out to match the statistics between  $|P|$  and  $|I|$  instead.

#### 4.2 Linear Power Spectrum Statistics

We first replace circular power spectrum statistics with linear statistics. Specifically, we *project* the 2D power spectrum onto a line  $l$  that corresponds to the motion direction in the spectral domain. We rotate the Fourier plane so that  $l$  is aligned with the  $u$  axis and apply the projection by integrating over  $v$ . This process can be alternatively viewed as applying a Radon Transform [25] along



**Fig. 3.** Power Spectrum Statistics on Five Randomly Selected Images from the Berkeley Segmentation Database [24]. (a) The circular power spectrum vs. the spatial frequency  $\omega$  in a log-log scale. The red lines show the fits of the  $1/\omega$ -exponent model. The scaling of the vertical axis belongs to the top trace. (b) The linear statistics along  $v$  vs.  $u$  in a log-log scale. The red curves show our estimated linear statistics from the circular statistics. For clarity, traces in both plots are shifted -1, -2, -3, and -4 log-units.

the  $v$  direction. In the discrete case, we can compute the linear averaged power spectrum of an image  $|J|$  as:

$$R_u[|J|] = \frac{1}{V} \sum_{v=0}^V |J(u, v)| \quad (15)$$

where  $V$  is the  $v$ -dimension resolution.  $R_u[|J|]$  represents the horizontal power spectrum statistics and can be approximated using Eq. (14) as:

$$R_u[|J|] \approx E\left[\frac{1}{V} \sum_{v=0}^V |J(u, v)|\right] = \frac{1}{V} \sum_{v=0}^V \frac{C}{(u^2 + v^2)^{m/2}} \quad (16)$$

Fig. 3(b) illustrates that our  $R_u[\cdot]$  estimation is accurate and robust.

We can further apply the  $R_u$  operator to both sides of Eq. (13):

$$R_u[|I|] = \sum_{v=0}^V |J(u, v)| |P(u)| = |P(u)| \cdot R_u[|J|] \quad (17)$$

Eq. (17) allows us to separate  $R_u[|J|]$  and  $|P|$ . We can further take the log of Eq. (17) as:

$$\log(R_u[|I|]) = \log(|P|) + \log(R_u[|J|]) \quad (18)$$

### 4.3 Motion Estimation

Fig. 4 illustrates our motion estimation algorithm. We first determine the motion direction and align it with the  $u$  axis. For every candidate motion parameter  $\alpha$ , we compute its FS-PSF  $p^\alpha$  and MTF  $|P^\alpha|$ , and use it to estimate the latent image power spectrum  $|J^\alpha| = |I|/|P^\alpha|$ . We then compute the linear statistics  $R_u[|J^\alpha|]$ , and  $R_u[|I|]$ . Finally, we compute the match score  $\mu$  between  $\log(|P^\alpha|)$  and  $\log(R_u[|I|]) - \log(R_u[|J^\alpha|])$ . The optimal motion parameter  $\alpha$  corresponds to the one that maximizes  $\mu$ .

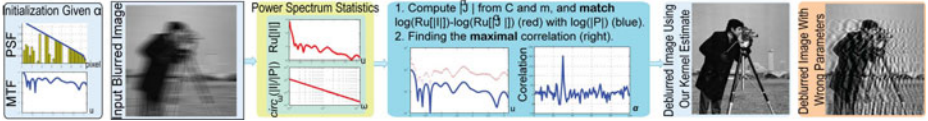


Fig. 4. Steps of Our Power-Spectrum-Based Motion Estimation Algorithm

**Estimating the Motion Direction.** We adopt a similar approach to [26] that finds the direction with most muted high frequencies. This assumes that the latent sharp image is not highly anisotropic, i.e., the power spectrum distribution along all directions have similar characteristics (variance, mean values). Since 1D motion blur attenuates the middle- and high-frequency information in the direction of motion, it amounts to a detection of a direction in which they are most muted. We do this by inspecting the Radon-power spectrum of the blurred image in all directions and choosing the one with the maximal variance.

**Computing Linear Statistics of  $|J^\alpha|$ .** A crucial step in our motion estimation algorithm is to derive the linear statistics of  $|J^\alpha| = |I|/|P^\alpha|$  from the circular statistics. Since we assume  $J^\alpha$  is motion blur free, its circular statistics should follow  $1/\omega$ -exponent distribution. To estimate  $C$  and  $m$ , we compute the discrete circular averaged power spectrum and apply line fitting between  $\log(\text{circ}_\omega[|J^\alpha|])$  and  $\log(\omega)$ . We then approximate the linear statistics  $R_u[|J^\alpha|]$  using Eq. (16).

**Matching Log-Linear Statistics.** Recall that our ultimate goal is to match  $f_1 = \log(|P^\alpha|)$  and  $f_2 = \log(R_u[|I|]) - \log(R_u[|J^\alpha|])$  under some metric  $\mu$ . A native  $\mu$  is to measure the squared difference at sampled points on  $f_1$  and  $f_2$ . Since the power spectrums of images generally have much smaller values in high frequency, directly computing the correlation between the estimate  $f_1$  and  $f_2$  results in unequal contributions from different frequencies.

We employ a metric based on the signs of the function derivatives to equally treat all frequencies. Specifically, we use a derivative sign function  $\Gamma(\cdot)$ :

$$\Gamma(\chi(u)) = \begin{cases} 1, & \frac{d\chi}{du} \geq 0 \\ -1, & \frac{d\chi}{du} < 0 \end{cases} \quad (19)$$

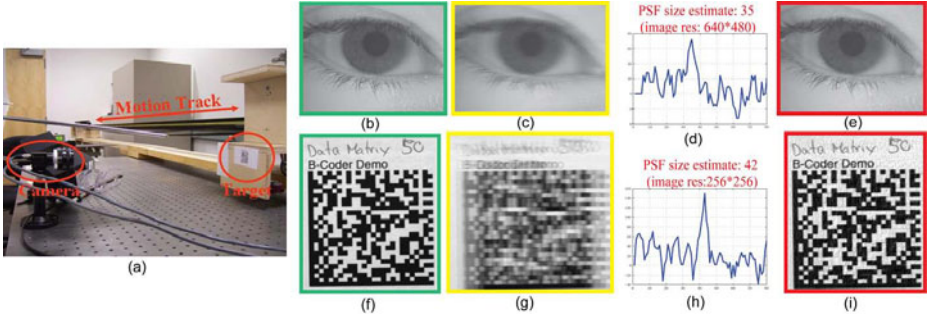
where  $\chi$  is a 1D function on  $u$ .

Finally, we sample  $f_1$  and  $f_2$  at discrete points  $u_1, u_2, \dots, u_n$ , and compute:

$$\mu(f_1, f_2) = \sum_{i=1}^n \Gamma(f_1(u_i)) \Gamma(f_2(u_i)) \quad (20)$$

#### 4.4 Motion-Aware Fluttered Shutter

Although not the focus of the paper, we briefly discuss how to use our techniques to develop motion-aware flutter shutters. The standard flutter shutter method has been focused on deblurring a single image. For videos, the object's motion



**Fig. 5.** Motion Estimation and Deblurring Results on an Iris Image and a Bar Code Image. (a) shows our motion stage and the fluttered shutter camera. Column 2: The ground truth blur-free images. Column 3: Blurred images caused by constant velocity motion under the FS. Column 4: The matching metric vs. the motion parameter (velocity). Column 5: The deblurred results using our recovered motion parameter.

may vary across the frames. Therefore, we aim to use the recovered motion to further update the initial shutter sequence to better match the motions.

Our strategy is to first determine the shutter sequence in the spatial domain and then map the sequence to the temporal domain. Recall that we have shown in Sec. 3 that the FS-PSF can be viewed as a motion envelope sampled by the shutter sequence: the envelope is a function of recovered motion parameters  $\alpha$  and the sampling is determined by the shutter sequence. We can directly model the FS-PSFs as a dot product of the envelope  $w(x)$  and a binary sequence  $b(x)$  in the spatial domain and apply the same search scheme in 1 and 5 to locate an optimal  $b(x)$  so that  $w(x)b(x)$  is "most" invertible, i.e., the one that has the maximal minimum magnitude in its MTF. Finally, we determine the flutter pattern  $s(t)$  from  $b(x)$  by using the motion model:

$$s(t) = s(t(x)) = b(x(t)) \quad (21)$$

Fig. 8 compares the deblur results using the initial const velocity optimal sequence and using our motion aware sequence.

## 5 Results

We have applied our technique to all of the publicly available flutter shutter images 1, and find that our method produces estimates that are within 1 pixel of the ground truth values, giving high-quality reconstructions. In order to test the broader types of blur (acceleration, harmonic motion) handled by our method, we have acquired additional test images using a Point Grey Flea2 camera triggered via the serial port of the controlling computer. The camera supports an external shutter mode that accumulates exposure over several chops, after which a single readout produces the flutter shutter image. To deblur the image from

our recovered FS-PSF, we use the linear system solution [1] for constant velocity motions and the Gaussian-derivative-prior method [21] for constant acceleration and harmonic rotation motions.

### 5.1 Constant Velocity

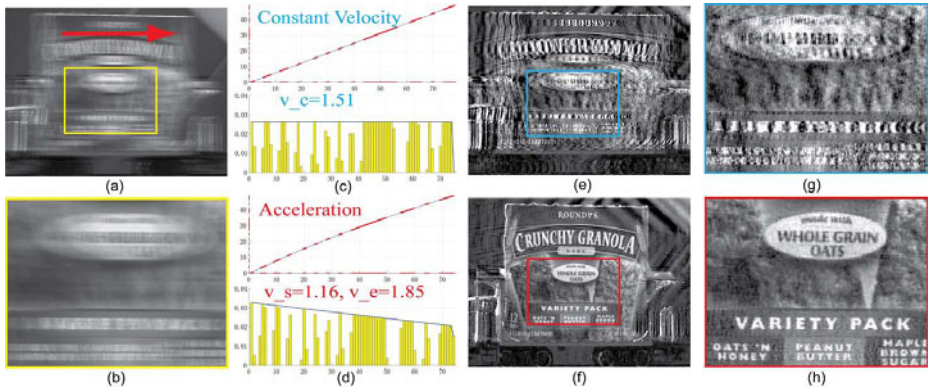
We first validate our algorithm on constant velocity motion. The only parameter here is the velocity. We captured the images from a fixed camera observing a motion stage to which textured objects are attached as shown in Fig. 5(a). Our motion stage can simulate different velocity motions via voltage controls. To measure the ground truth velocity, we use a step edge calibration target and measure its blurred width in an image with a known exposure time. We choose the shutter sequence as in [1] whose MTF has the maximum min magnitude and has a chop duration of  $v_c = 1\text{pixel}/\text{chop}$ . With this setup, we obtain the ground truth FS-PSF using Eq. (5).

Fig. 5 shows two examples acquired using this setup, an iris image (b) with little texture and a bar code image (f) with repetitive texture. The iris and 2D barcode targets move from left to right with a constant velocity, giving the flutter shutter images (c) and (g). In both cases the motion is axis-aligned horizontal. Our estimated motion direction is within  $1^\circ$  degree of this ground truth. The plots (d) and (h) show the matching metric  $\mu$  computed over a range of potential PSF sizes (proportional to velocities in this case), which have pronounced peaks exactly at the ground truth values (35 pixels for the iris image and 42 pixels for the barcode image). The resulting FS-PSF estimates are then used to deblur (c) and (g). Our deblurred results (e) and (i) contain sufficient detail to perform recognition on the de-blurred images. The iris template extracted from our de-blurred image was successfully matched to a separate image of the same eye, and the barcode image can be decoded to extract its payload. Neither the iris recognition nor the barcode decoding were successful on Lucy-Richardson [27] de-blurred versions of traditional shutter images captured with the same setup.

### 5.2 Constant Acceleration

For constant acceleration, the motion parameters  $\alpha$  are the starting velocity and acceleration. We capture accelerated motion images using a toy car on a slanted track, using a dead drop for which gravity provides the only acceleration. Because of the unknown timing between the release of the car and the image capture, we are unable to determine the ground truth FS-PSF for these images. Instead, we validate our motion estimation by the quality of the deblurred results. The shutter sequence used in these experiments is computed under the constant velocity assumption and does not account for acceleration. Because the velocity and acceleration are unknown a priori, these images are generated with what is essentially a random flutter shutter sequence.

Fig. 6 (a) shows the image captured as the toy train undergoes accelerated motion. Though the track is slanted at  $55^\circ$ , the camera is rotated so that the motion appears nearly horizontal. We first apply our motion direction estimation



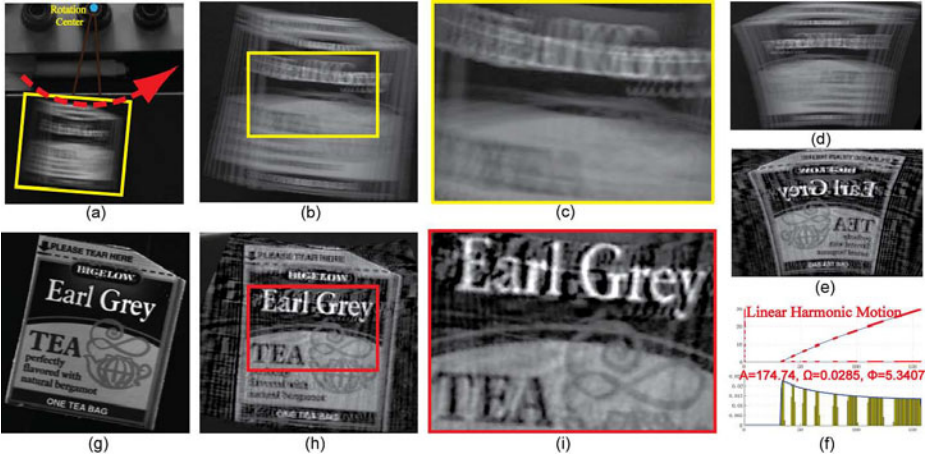
**Fig. 6.** FS-PSF Estimation and Deblurring Results on Constant Accelerations. A toy car sliding down a slanted track at  $55^\circ$ . The camera is rotated so that the motion appears horizontal. For clarity, a textured cardboard was attached to the car. (a): The motion blurred image under the fluttered shutter. (c) and (d) show the functions  $t(x)$  and the corresponding PSFs using constant velocity assumption and our recovered constant acceleration motion. (e) The deblurred result using the FS-PSF of constant velocity ( $v_c=1.51$ pixels/chop). (f) The deblurred result using our algorithm’s FS-PSF estimation for constant acceleration ( $v_s=1.16$ pixel/chop,  $v_e=1.85$ pixel/chop). (b), (g), and (h) are close-up views for (a), (e), and (f).

algorithm, which produces an estimate of  $1^\circ$ . Next, we apply our power spectrum statistics approach to determine the acceleration motion parameter, which gives  $t(x)$  and the FS-PSF shown in (d). The deblurred result is shown in (f), and a close-up in (h). Given the severe blur in (a) and the fact that the fluttering sequence is not optimal under accelerated motion, the amount of detail present in the close-up is significant. Note that reconstruction artifacts in (f) are due to the stationary background’s intensity interacting with the moving foreground object. We also present the deblurred result assuming a constant velocity motion model. Our algorithm first estimates the motion velocity and plots the  $t(x)$  and PSF in (c). As shown in (e) and (g), using incorrect motion model, the deblurred images contain severe artifacts.

### 5.3 Harmonic Rotation

Finally, we experiment our approach on planar harmonic rotation. The harmonic rotation consists of 3 parameters, i.e.,  $A$ ,  $\Omega$ , and  $\Phi$ . As shown in Fig. 7(a), we emulate harmonic rotation by hanging a heavy rigid object below a fixed stick using two approximately rigid, weightless cords. These two cords are connected to the same point. By swinging the object back and forth freely within a plane, we synthesize a periodic harmonic rotation. Notice that the rotation is 2-dimensional, with spatially varying blur kernels for different pixels (Fig. 7(b)).

In order to simplify the analysis, we transform the harmonic rotation into a linear harmonic motion. Specifically, we track feature points and estimate the



**Fig. 7.** FS-PSF Estimation and Deblurring Results on a Harmonic Rotational Tea Bag (g). (a) and (b): The captured blur image under the FS. (d): Warped (b) under polar coordinates. (e) and (h): Our motion deblurred result under the polar and the cartesian coordinates. (f): Our recovered FS-PSF. (c) and (i): Close-up views for (b) and (h).

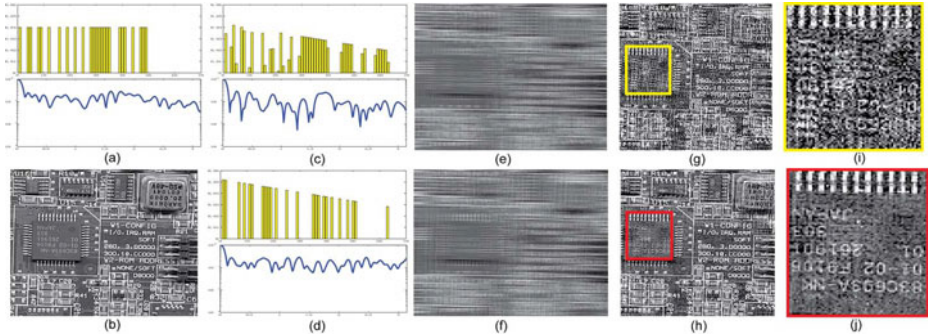
rotation center by solving a least squares problem [16]. We then warp the image along the radial directions to form a spatially invariant linear harmonic motion blur (d). Our algorithm recovers the harmonic motion parameters (f) and then deblurs the image (d) and obtain (e). Finally, we warp the image back to the original cartesian coordinate system (h).

#### 5.4 Motion Aware Shutter Sequence

We pick a fluttered shutter sequence originally designed optimal for constant velocity motion ( $v=1.0\text{pixel}/\text{chop}$ ) as the initial shutter sequence. We then use this sequence to capture an accelerated motion with  $v_s=0.8$  pixels/chop and  $v_e=1.8\text{pixels}/\text{chop}$ . The resulting FS-PSF is shown in Fig. 8(c). Notice that it has small values at several frequencies. We synthetically blur a sharp image (b) using the FS-PSF with additive Gaussian white noise of  $\sigma=0.01$ . We then deblur it using our motion estimation algorithm, with constant acceleration. Although our method recovers highly accurate motion parameters ( $v_s=0.790$ ,  $v_e=1.805$ ), the resulting deblurred results contain strong ringing artifacts.

Using the recovered motion parameter, we apply the random search scheme as in [1] to find the optimal flutter shutter sequence. The new FS-PSF is shown in Fig. 8(d). Compared with the old FS-PSF, it maintains large values at all frequencies. We use the new FS-PSF to blur the sharp image and also add Gaussian white noise  $\sigma=0.01$ . Finally, we apply our power spectrum statistics method to recover the motion parameter and obtain a new deblurred image as shown in (h). (i) and (j) show the close-up views of the deblurred results under the old and new FS-PSF. The motion-aware FS-PSF yields much less artifacts.





**Fig. 8.** Motion-aware Fluttered Shutter. (a) is the PSF and the MTF under constant velocity. (c) is the PSF/MTF under acceleration motion ( $v_s=0.8$ pixels/chop,  $v_e=1.8$ pixels/chop) using the same sequence as in (a). (d) is the PSF/MTF using our motion-aware sequence. (e) and (f) are synthetically blurred images of (b) using acceleration PSFs in (c) and (d). (g) and (h) are the corresponding deblurred images, (i) and (j) show the close-up views.

## 6 Conclusion and Limitations

We have presented a new fluttered-shutter-based motion estimation and deblurring framework. Our method adopts the fluttered-shutter point-spread-function (FS-PSF) model to uniformly describe blur kernels under general motions. We have developed an automatic motion-from-blur technique that recovers the FS-PSF by analyzing image power spectrum statistics. We have introduced a new linear statistics model that can be directly estimated from circular power spectrum statistics. We have shown that the MTF of 1D FS-PSF should be statistically correlated to the linear statistics of the blurred image’s power spectrum along the motion directions. To find the optimal FS-PSF, our method searches the space of motion parameters to find the one that yields the maximum correlation.

The use of fluttered shutters is crucial in our motion-from-blur algorithm. Recall that the first step in our linear statistics estimation is to compute the latent image power spectrum  $|J| = |I|/|P|$ . The implicit assumption there is that  $|P|$  does not contain zeros, the most important property of the fluttered shutter. For conventional shutters where  $P$  is a sinc function and has many zeros, the resulting  $|J|$  will contain points with large values and robustly fitting  $1/\omega$ -exponent distribution to circular power spectrum statistics is difficult. Thus, our technique is not directly applicable to the box filters.

Another limitation of our framework is that it is restricted to 1D motions. 1D motions allows us to efficiently separate the FS-PSF from linear statistics of the latent image (Eq. (17)). Intuitively, our technique may be directly applied to 2D motions. For example, once we compute  $C$  and  $m$  of the  $1/\omega$ -exponent model, we can approximate  $|J(u, v)| \approx C/(u^2+v^2)^{m/2}$  and directly match the 2D function  $|I|/|J|$  with  $|P|$ . However, since the  $1/\omega$ -exponent model is a statistical

model, the actual  $|J|$  values may significantly deviate from their expected values. Therefore, matching 2D  $|I|/|J|$  with  $|P|$  is not reliable. A possible solution for future work is to approximate the 2D FS-PSF as combinations of 1D FS-PSFs and then reapply our linear statistics method to fit along their corresponding directions. Another important future direction is to use our motion-aware flutter shutter for video deblurring. The challenge there is to determine the optimal shutter sequence from the estimated motions in real-time (e.g., 30fps). Recall that majority of our computations lie in the spectral space and computing image statistics is similar to texture filtering. Therefore, we plan to re-implement our algorithm on the GPU for real-time motion estimation and shutter selection.

## Acknowledgement

The authors would like to thank the reviewers for their insightful comments. Part of this work was done while the first author was an intern at Honeywell Labs. Yuanyuan Ding and Jingyi Yu were partially supported by NSF grants MSPA-MCS-0625931 and IIS-CAREER-0845268.

## References

1. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.* 25, 795–804 (2006)
2. Yitzhaky, Y., Mor, I., Lantzman, A., Kopeika, N.S.: Direct method for restoration of motion-blurred images. *Journal of the Optical Society of America A: Optics, Image Science, and Vision* 15, 1512–1519 (1998)
3. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: *NIPS*, pp. 1–8 (2009)
4. Levin, A.: Blind motion deblurring using image statistics. In: *NIPS*, pp. 841–848 (2007)
5. Agrawal, A., Xu, Y.: Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In: *CVPR* (2009)
6. van der Schaaf, A., van Hateren, J.H.: Modelling the power spectra of natural images: Statistics and information. *Vision Research* 36, 2759–2770 (1996)
7. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.* 26, 1 (2007)
8. Ott, K.A., Kraemer, F., Lin, Y., McAdoo, B., Wang, J., Widemann, D., Wohlberg, B.: Blind image deconvolution: Motion blur estimation. In: *Technical Report for the Mathematical Modeling in Industry X Workshop* (2006)
9. Moghaddam, M.E., Jamzad, M.: Fining point spread function of motion blur using radon transformation and modeling the motion length. In: *ISSPIT* (2004)
10. Ji, H., Liu, C.: Motion blur identification from image gradients. In: *CVPR* (2008)
11. Dai, S., Wu, Y.: Motion from blur. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
12. Haykin, S.: *Blind Deconvolution*. Prentice-Hall, Englewood Cliffs (1994)
13. Jia, J.: Single image motion deblurring using transparency. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07*, pp. 1–8 (2007)

14. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. In: SIGGRAPH '08, pp. 1–10 (2008)
15. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. In: SIGGRAPH '06, pp. 787–794 (2006)
16. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: ICCV 2007, pp. 1–8 (2007)
17. Agrawal, A., Xu, Y., Raskar, R.: Invertible motion blur in video. *ACM Trans. Graph.* 28, 1–8 (2009)
18. Ben-Ezra, M., Nayar, S.K.: Motion-based motion deblurring. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 689–698 (2004)
19. Li, F., Yu, J., Chai, J.: A hybrid camera for motion deblurring and depth map super-resolution. In: CVPR, pp. 1–8 (2008)
20. Tai, Y.W., Du, H., Brown, M.S., Lin, S.: Image/video deblurring using a hybrid camera. In: CVPR, pp. 1–8 (2008)
21. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. In: ACM SIGGRAPH (2007)
22. Joshi, N., Zitnick, C., Szeliski, R., Kriegman, D.: Image deblurring and denoising using color priors, pp. 1550–1557 (2009)
23. Favaro, P., Soatto, S.: Learning shape from defocus. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2351, pp. 735–745. Springer, Heidelberg (2002)
24. Martin, D.R., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB/CSD-01-1133, EECS Department, University of California, Berkeley (2001)
25. Toft, P.: The Radon Transform — Theory and Implementation. PhD thesis, Electronics Institute, Technical University of Denmark, Lyngby, Denmark (1996)
26. Oliveira, J.P., Figueiredo, M.A., Bioucas-Dias, J.M.: Blind estimation of motion blur parameters for image deconvolution. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) *IbPRIA 2007*. LNCS, vol. 4478, pp. 604–611. Springer, Heidelberg (2007)
27. Lucy, L.B.: An iterative technique for the rectification of observed distributions. *Astron. Journal* 79, 745 (1974)

# Error-Tolerant Image Compositing

Michael W. Tao<sup>1</sup>, Micah K. Johnson<sup>2</sup>, and Sylvain Paris<sup>3</sup>

<sup>1</sup> University of California, Berkeley

<sup>2</sup> Massachusetts Institute of Technology

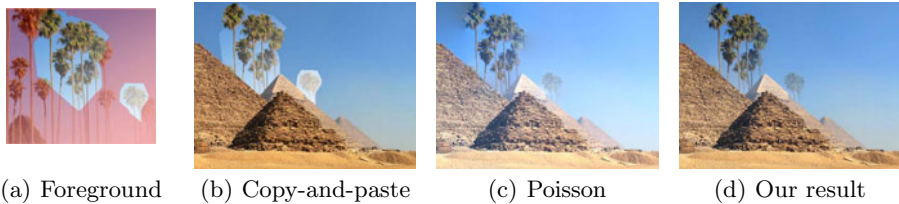
<sup>3</sup> Adobe Systems, Inc.

**Abstract.** Gradient-domain compositing is an essential tool in computer vision and its applications, e.g., seamless cloning, panorama stitching, shadow removal, scene completion and reshuffling. While easy to implement, these gradient-domain techniques often generate bleeding artifacts where the composited image regions do not match. One option is to modify the region boundary to minimize such mismatches. However, this option may not always be sufficient or applicable, e.g., the user or algorithm may not allow the selection to be altered. We propose a new approach to gradient-domain compositing that is robust to inaccuracies and prevents color bleeding without changing the boundary location. Our approach improves standard gradient-domain compositing in two ways. First, we define the boundary gradients such that the produced gradient field is nearly integrable. Second, we control the integration process to concentrate residuals where they are less conspicuous. We show that our approach can be formulated as a standard least-squares problem that can be solved with a sparse linear system akin to the classical Poisson equation. We demonstrate results on a variety of scenes. The visual quality and run-time complexity compares favorably to other approaches.

**Keywords:** gradient-domain compositing, visual masking.

## 1 Introduction

Gradient-domain compositing is an essential technique at the core of many computer vision applications such as seamless cloning [1–4], panorama stitching [5–7], inpainting [8], shadow removal [9], scene completion [10], and reshuffling [11]. These methods first delineate the composited regions, then compute a target gradient field and boundary conditions from these regions, and finally solve the Poisson equation to reconstruct an image. A major issue with gradient-domain compositing is that the combined gradient field may not be integrable; that is, an image with gradients that match the target field as well as the specified boundary conditions may not exist. Existing work mitigates this issue by moving the boundary to more carefully combine the merged regions. However, when the combined images are widely different, this strategy may not be sufficient. Or, if the user has specified the boundary by hand, he or she may not want it to be altered. For instance in Figure 1, the selection cannot be modified



**Fig. 1.** We present an image-compositing technique tolerant to selection inaccuracies. In this example, a user wishes to add trees to an image of the Egyptian pyramids, but it is not possible to select the trees without cutting through the foliage (a). Moreover, to ensure a good insertion behind the pyramids, it is not possible to modify the selection boundary. A direct copy of the pixels yields a undesirable visible seam (b). Standard gradient-domain compositing minimizes the seam, but leads to bleeding artifacts where the foliage is cut (c). Our method characterizes where color leakage should be avoided, producing a seamless composite without bleeding artifacts (d).

because the tree trunks have to abut the pyramids. Even with boundary refinement, the target gradient fields may be far from integrable, yielding color leaks and halos typical of Poisson-based methods.

In this paper, we present an algorithm for minimizing artifacts in gradient-domain image compositing. We characterize the origin of typical bleeding artifacts and analyze the image to locate the areas where they would be most and least conspicuous. Based on this analysis, we propose a two-step algorithm. First, we process the gradient values on the boundary to minimize artifacts in regions where bleeding would be visible. Second, we describe a weighted integration scheme that reconstructs the image from its gradient field so that residuals are located in textured regions where they are less visible. Our results show that the combination of these two steps yields significantly better composites. Moreover, our method is formulated as a least-squares optimization that can be solved using a sparse linear system, which makes our approach computationally efficient. We demonstrate our approach on scenarios in which boundary mismatches are likely to occur: user-driven seamless cloning [1], heterogeneous panorama stitching [7], and scene reshuffling [11].

## 1.1 Related Work

Gradient-domain techniques are useful to a variety of problems in computer vision, including image stitching, intrinsic images, shadow removal, and shape-from-shading [5, 12–15]. In most of these problems, the gradient field contains non-integrable regions and many authors have noted that reconstruction artifacts are often due to boundary conditions. As a result, a variety of methods have been introduced to minimize artifacts by refining the boundary location [2, 4, 5, 16]. Rather than moving the boundary, which may not always be possible, we focus on reconstructing the final image from the target gradient field once the boundary is specified. Our approach is complementary and orthogonal to boundary-refinement methods. We show that our image analysis combined with a careful study of the numerical scheme reduces visible

artifacts. Our approach could benefit many computer vision algorithms that rely on gradient-domain reconstruction as a subroutine.

The general formulation of the gradient-domain reconstruction problem is to seek an image  $I$  that approximates the target field  $\mathbf{v}$  in a least-squares sense (with  $\nabla$ , the gradient operator):

$$\operatorname{argmin}_I \int \|\nabla I - \mathbf{v}\|^2 \quad (1)$$

which can be minimized by solving the Poisson equation:

$$\Delta I - \operatorname{div}(\mathbf{v}) = 0 \quad (2)$$

where  $\Delta$  is the Laplacian operator  $\partial^2/\partial x^2 + \partial^2/\partial y^2$  and  $\operatorname{div}$  is the divergence operator  $\partial/\partial x + \partial/\partial y$ . To solve this equation, one also needs boundary conditions that depend on the application. We illustrate how to compute the target gradient  $\mathbf{v}$  in the context of seamless compositing using three inputs: the background image,  $B$ ; the foreground image,  $F$ ; and a selection,  $\mathcal{S}$  with a boundary  $\beta$  [5].

$$\mathbf{v}(x, y) = \begin{cases} \nabla F & \text{if } (x, y) \in \mathcal{S}, (x, y) \notin \beta \\ \nabla B & \text{if } (x, y) \notin \mathcal{S} \\ \frac{1}{2}(\nabla F + \nabla B) & \text{if } (x, y) \in \beta \end{cases} \quad (3)$$

Other cases such as panorama stitching are similar except that the images are not named “foreground” and “background.” For the sake of simplicity, we will name the images foreground and background.

The gradients from the foreground image  $F$  and background image  $B$  are integrable since they are computed directly from images. But the gradients along the boundary between the two images may not be integrable, creating a source of errors that the integration routine must manage. Farbman et al. [17] address this issue by relying on users to identify the leaks. The gradients of marked regions are ignored, which removes the leaks. In comparison, our method analyzes the image to automatically adapt the integration process. Our approach shares similarities with the method of Lalonde et al. [16] who propose to take the image gradient magnitude into account during the reconstruction process. However, color leaks may still appear with this technique when boundaries are not accurate.

Besides image compositing, gradient-domain methods have also been used in computer vision for surface reconstruction problems, such as shape-from-shading and photometric stereo. In these problems, an algorithm estimates the gradient of a surface at every pixel and then a robust Poisson solver is used to find the surface that best fits the estimated gradients. We refer to the recent work of Agrawal et al. [14], Reddy et al. [18], and the references therein for detail. Although image compositing and robust integration techniques both reconstruct a 2D signal from its gradients, the two problems are fundamentally different. The gradients from surface-reconstruction methods are noisy everywhere, whereas image-compositing gradients are problematic only at the boundary between foreground and background. In this paper, we exploit this specificity to improve the quality of the results. We also rely on visual masking to locate integration residuals where they are less conspicuous.

## 1.2 Contributions

In this paper, we introduce several contributions.

▷ *Low-curl boundaries.* We describe a method that limits the artifacts by minimizing the curl of the target gradients on the foreground-background boundary.

▷ *Weighted Poisson equation.* We show how to add weights to the Poisson equation so that integration residuals lie in textured regions where they are less visible due to visual masking.

▷ *Efficient non-bleeding compositing.* We combine the two previous contributions to obtain a compositing algorithm that prevents bleeding artifacts while remaining linear akin to the original Poisson equation.

## 1.3 Overview

Our algorithm consists of two steps. First, we focus on the boundary between the foreground and background regions. We characterize the origin of the bleeding artifacts and we show how to modify the gradient field  $\mathbf{v}$  to minimize them. The second step focuses on the location of the integration residuals. We show that artifacts are less visible in textured regions due to visual masking. We describe an algorithm that controls the integration residuals such that they are located in textured areas. In the results section, we show that the combination of these two steps yields visually superior results.

## 2 Low-Curl Boundary

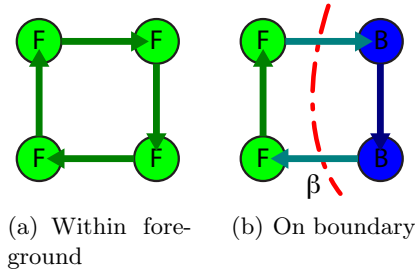
A necessary condition for a gradient field  $\mathbf{u}$  to be integrable is to have a zero curl<sup>1</sup>. That is, if there exists an image  $I$  such that  $\nabla I = \mathbf{u}$ , then  $\text{curl}(\mathbf{u}) = \partial \mathbf{u}_y / \partial x - \partial \mathbf{u}_x / \partial y = 0$ . For example, consider the configuration illustrated in Figure 2(a). When all pixels come from one image, in this case the foreground image, the derivatives are consistent and the curl is zero. Therefore, this region is integrable, i.e., the image can be reconstructed from its gradients. The same observation holds for regions from the background image.

In the image compositing problem, a non-integrable gradient field only occurs on the boundary, as illustrated in Figure 2(b). On the boundary, the gradient field is a mixture of two fields and may have non-zero curl since gradients come from mixed sources. When the gradient field has a non-zero curl, we cannot minimize the Poisson equation (2) exactly and residuals remain. These residuals are often visible in composited images as halos, bleeding, or other artifacts.

### 2.1 Reducing the Curl on the Boundary

Since the non-integrability of regions along boundary is the source of artifacts, we seek to alter the desired gradient field to minimize the bleeding artifacts. Let  $\mathbf{v}$  represent the desired gradient field of the composited image. To preserve

<sup>1</sup> Note that for a 2D vector field  $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y)$ , the curl is a scalar value that corresponds to the  $z$  component of the 3D curl applied to the 3D vector field  $(\mathbf{u}_x, \mathbf{u}_y, 0)$ .



**Fig. 2.** *Estimating the curl on a discrete grid.* Circles denote pixels and arrows denote differences between pixels. If the curl is computed within the foreground region (a), all the derivatives come from  $F$  and the curl is null. The background case is equivalent (not shown). On the boundary (b), derivatives from diverse sources are used and in general the curl is not zero.

the visual information of the two images, we do not modify the foreground or background gradients in  $\mathbf{v}$ . We only modify  $\mathbf{v}$  values on the boundary such that the curl is as small as possible.

A naive solution would be to seek  $\text{curl}(\mathbf{v}) = 0$  everywhere. But the following counterexample shows that this approach would not achieve our goal. Consider the standard copy-and-paste operation that directly combines the pixel values and produces an image  $I_{\text{seam}}$  with visible seams. The curl of the gradient field of  $I_{\text{seam}}$  is null since it is computed from an actual image. And, inside the selection, gradients are equal to the foreground values since pixels have been copied. The same holds outside the selection with the background values. However, on the boundary, gradients are different from either the foreground or the background, which generates the seams. We address the shortcomings of this naive solution by seeking gradient values that minimize the curl and are close to the gradients of the input images.

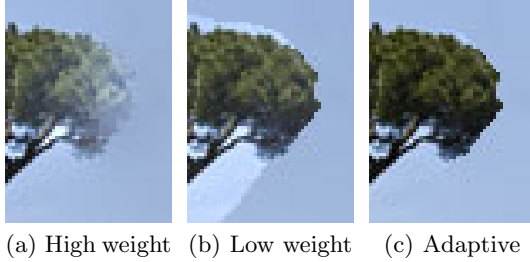
*A Least-squares Approach.* We formulate our goal using a least-squares energy where the desired gradients  $\mathbf{v}$  on the boundary are the unknowns. The first term minimizes the curl:  $\int_{\beta} [\text{curl}(\mathbf{v})]^2$  and the second term keeps the values close to the input gradients  $\int_{\beta} (\mathbf{v} - \nabla F)^2 + \int_{\beta} (\mathbf{v} - \nabla B)^2$ . This last term has the same effect as keeping  $\mathbf{v}$  close to the average gradient. We combine the two terms to obtain:

$$\operatorname{argmin}_{\mathbf{v}} \int_{\beta} \left( [\text{curl}(\mathbf{v})]^2 + W_{\beta} \left[ \mathbf{v} - \frac{1}{2}(\nabla B + \nabla F) \right]^2 \right) \quad (4)$$

where  $W_{\beta}$  controls the importance of the second term.

*Adaptive Weights.* Figure 3 shows results for several values of  $W_{\beta}$ . For large  $W_{\beta}$ , we only minimize the proximity to the input gradients, which is the standard gradient compositing with seamless boundaries but leaking artifacts. For a small  $W_{\beta}$ , we have the naive solution described above where we only minimize the curl. There are no bleeding artifacts but the boundary is visible. We combine these two behaviors by varying the weights according to the local image structure.





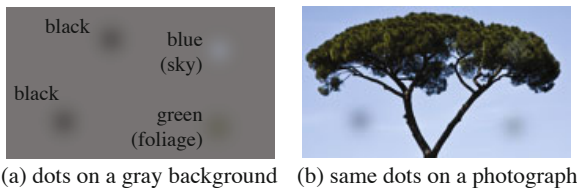
**Fig. 3.** *Influence of the curl term.* With high weights  $W_\beta$ , the composite is seamless but suffers from bleeding (a). With low  $W_\beta$ , bleeding disappears but seams become visible (b). Our adaptive approach locally adjusts the weights to achieve seamless results with no leaks (c).

Intuitively, a seamless boundary is desirable when both sides of the boundary are smooth. This is the case for instance when we stitch a sky region with another sky region. A seamless boundary is also acceptable when both sides are textured because leaking is a low-frequency phenomenon that will be hidden by visual masking. Figure 4 illustrates this effect that has also been used in the rendering literature [19–22]. In these two cases, we seek high values for  $W_\beta$ . But when a textured region is composited adjacent to a smooth region, we want to prevent bleeding because such regions would generate unpleasing artifacts on the smooth side, e.g. in the sky. In this case, we want low values of  $W_\beta$ . The following paragraph explains how we compute  $W_\beta$  based on the local amount of texture.

*Estimating the Local Amount of Texture.* Our strategy relies on the presence or absence of texture in a given neighborhood. In this paragraph, we describe a simple and computationally efficient texture estimator although one could use other models [23, 24]. Formally, our scheme is:

$$T_{\sigma_1, \sigma_2}(\mathbf{g}) = \frac{G_{\sigma_1} \otimes \|\mathbf{g}\|}{G_{\sigma_2} \otimes \|\mathbf{g}\|} n(\|\mathbf{g}\|) \quad (5)$$

where  $\mathbf{g}$  is a gradient field,  $G_\sigma$  is a Gaussian of width  $\sigma$ ,  $\sigma_1$  and  $\sigma_2$  are two parameters such that  $\sigma_1 < \sigma_2$ ,  $\otimes$  is the convolution operator, and  $n(\cdot)$  a noise-controlling function. Our scheme relies on image gradients, for instance  $T(\nabla I)$  is the texture map of the image  $I$ . We compare the average amplitude of the



**Fig. 4.** We show the same dots on a uniform background (a) and on a photograph (b) but the two dots on the tree are not visible because of the texture of the foliage

gradients in two neighborhoods defined by  $\sigma_1$  and  $\sigma_2$ . If the image is locally textured, then the average in the small neighborhood will be higher than in the large neighborhood, corresponding to  $T > 1$ . Conversely,  $T < 1$  corresponds to regions with locally less texture than in the larger neighborhood. This scheme would be sensitive to noise in smooth regions where gradients are mostly due to noise. We address this issue with the function  $n$  that is equal to 0 for very small gradient and 1 otherwise. In practice, we use a smooth step equal to 0 for the bottom 2% of the intensity scale and 1 for 4% and above. In our context, the goal is to differentiate textureless areas from textured regions; the relative amount of texture in textured regions does not matter. Consequently, we use the estimator  $\bar{T} = \min(1, T)$  that considers all textured regions to be equal.

*Computing the Boundary Weights.* Recall that we want  $W_\beta$  to be large when both foreground and background regions have the same degree of texture, either both smooth or both textured. If one of them is smooth and the other textured, we want  $W_\beta$  to be small. We consider the difference  $D = |\bar{T}(\nabla F) - \bar{T}(\nabla B)|$  and define  $W_\beta$  using a function that assigns a small value  $w$  when  $D$  is large, 1 when  $D$  is small, and linearly transitions between both values. Formally, we use:

$$W_\beta = \begin{cases} w & \text{if } D > \tau \\ \min(1, w + \lambda(1 - D/\tau)) & \text{otherwise} \end{cases} \quad (6)$$

where  $\lambda$  and  $\tau$  control the linear transition. We found that  $\lambda = 4$ ,  $w = 0.05$ ,  $\tau = 0.005$ ,  $\sigma_1 = 0.5$ , and  $\sigma_2 = 2$  work well in practice. All results are computed with these values unless otherwise specified.

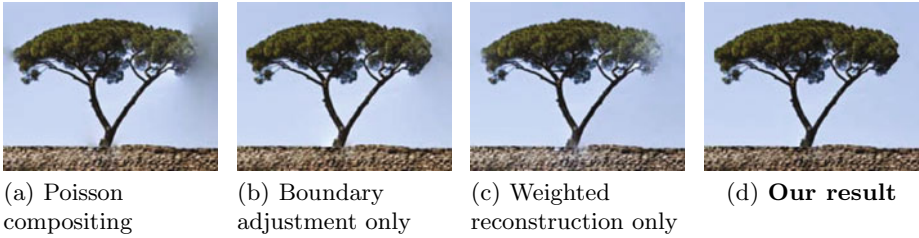
*Discussion.* Figure 5 illustrates the effect of our approach that reduces the curl on the compositing boundary. Bleeding artifacts are significantly reduced. In next section, we describe how to remove the remaining leaks. For color images, we use RGB gradients in Equation (5) so that we account for luminance and chrominance textures. From an efficiency standpoint, an important characteristic of our approach is that it can be solved with a sparse linear system since our least-squares energy (Eq. 4) involves only sparse linear operators and  $W_\beta$  depends only on the input data.

### 3 Controlling the Location of the Residuals

Although our boundary treatment reduces the curl of the gradient field  $\mathbf{v}$ , in general  $\mathbf{v}$  is not integrable. As with other gradient-domain methods, our goal is to produce an image with a gradient field  $\nabla I$  as close as possible to  $\mathbf{v}$ . Our strategy is to modify the Poisson equation (Eq. 2) in order to locate the residuals as much as possible in regions where they will be the least objectionable. Intuitively, we want to avoid errors in smooth regions such as the sky where they produce color leaks and halos, and put them in textured areas where visual masking will conceal the artifacts (Fig. 4).

#### 3.1 Adapting the Poisson Equation

Let's assume that we have a scalar map  $W_P$  with high values in regions where errors would be visible and low values otherwise. We discuss later how to compute



**Fig. 5.** To reduce the color bleeding artifacts visible in a Poisson composite (a), we proceed in two steps. We adjust the gradient values at the boundaries to minimize the curl and weight the reconstruction process so that residual is mostly concentrated in the textured regions. While these two steps improve the results when applied separately (b,c), combining them achieves a visually superior composite (d).

such a function using our texture estimator  $\bar{T}$ . Given  $W_P$ , we modulate the least-squares strength so that we penalize less the regions where we prefer the residuals to be, that is, regions with low  $W_P$  values:

$$\operatorname{argmin}_I \int W_P \|\nabla I - \mathbf{v}\|^2 \quad (7)$$

Since we want to reduce the difference between  $\nabla I$  and  $\mathbf{v}$ ,  $W_P$  has to be strictly positive everywhere. Moreover, to keep our approach computationally efficient, we will design  $W_P$  such that it does not depend on the unknown image  $I$ . In this case, Equation 7 is a classical least-squares functional that can be minimized by solving a linear system. To obtain a formula similar to the Poisson equation (2), we apply the Euler-Lagrange formula [25]. Recall that  $W_P$  does not depend on  $I$ . Thus, we obtain the following linear system:

$$\operatorname{div}(W_P(\nabla I - \mathbf{v})) = 0 \quad (8)$$

In Section 3.2, we show that although this equation is simple, it has favorable properties.

*Computing the Weights.* To keep our scheme linear, we do not use any quantity related to the unknown  $I$ . We use the desired gradient field  $\mathbf{v}$  to estimate the texture location in the image. Although  $\mathbf{v}$  is not equal to the gradient of final output, it is a good approximation that is sufficient to compute the weights  $W_P$ . Since we want high weights in smooth regions and low weights in textured areas, we use the following formula:  $W_P = 1 - p \bar{T}(\mathbf{v})$  where  $p$  is a global parameter that indicates how much we control the residual location. For instance,  $p = 0$  corresponds to no control, that is, to the standard Poisson equation, whereas larger values impose more control.  $p$  has to be strictly smaller than 1 to keep  $W_P > 0$ . We found that values close to 1 performs better in practice. We use  $p = 0.999$  in all our results. We also found that it is useful to have a more local estimate of the texture, which we achieve using  $\sigma_1 = 0$  to compute  $\bar{T}$  while keeping the other parameters unchanged.

### 3.2 Analysis of the Residual Structure

Independent of the actual definition of  $W_P$ , we can show that the residuals produced by our approach have structure that is aligned with the image content. Wang et al. [26] have demonstrated that such structural similarity produces more acceptable results. To better understand the role of  $W_P$ , we distribute the divergence in Equation 8:  $W_P \operatorname{div}(\nabla I - \mathbf{v}) + \nabla W_P \cdot (\nabla I - \mathbf{v}) = 0$ . With  $W_P \neq 0$ , the relation  $\operatorname{div}(\nabla I) = \Delta I$ , and the logarithmic gradient  $\nabla W_P / W_P = \nabla \log W_P$ , we obtain:

$$\underbrace{\Delta I - \operatorname{div}(\mathbf{v})}_{\text{Poisson term}} + \underbrace{\nabla \log W_P \cdot (\nabla I - \mathbf{v})}_{\text{new term}} = 0 \quad (9)$$

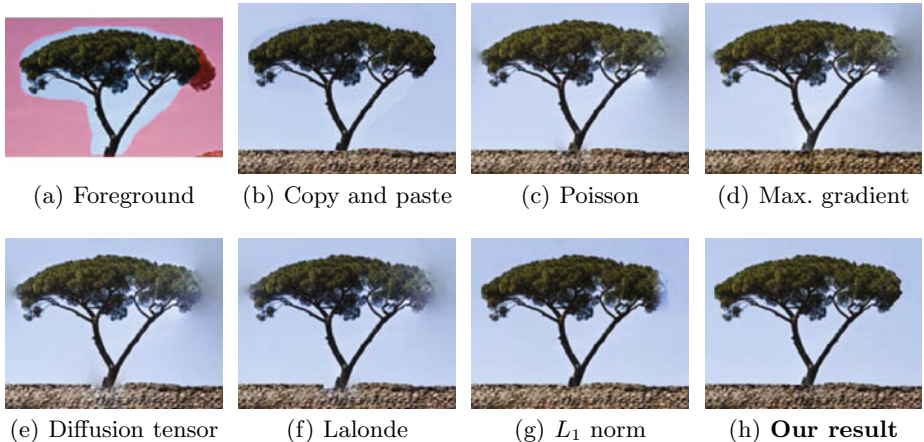
The left term is the same as the standard Poisson equation (2) while the right term is new. In regions where  $W_P$  is constant, the new term is null and our scheme behaves as the Poisson equation, that is, it spreads the residuals uniformly. In other regions where  $W_P$  varies, our scheme differs from the Poisson equation and allows for discontinuities in the residual. Since  $W_P$  measures the amount of texture, it means that residual variations are aligned with texture edges, which ensures the structural similarity that has been shown desirable by Wang et al. [26]. We provide illustrations of this property in supplemental material.

### 3.3 Relationship with Existing Methods

For this section, we make explicit the variable  $W_P$ , that is, Equation 7 becomes  $\int W_P(\mathbf{v}) \|\nabla I - \mathbf{v}\|^2$ , and Equation 8,  $\operatorname{div}(W_P(\mathbf{v}) (\nabla I - \mathbf{v})) = 0$ . We discuss the relationships among our work and related methods independently of the actual definition of  $W_P$ .

*The Poisson Equation and its Variants.* Rewriting the Poisson equation (2) as  $\operatorname{div}(\nabla I - \mathbf{v}) = 0$ , we see that our linear system has the same complexity since we do not introduce new unknowns nor new coefficients in the system; we only reweight the coefficients. Agrawal et al. [14] also describe an anisotropic variant that is linear. However, while this method performs well in shape-from-shading, it does not prevent bleeding when applied to image compositing (Fig. 6). The  $L_1$  reconstruction method that Reddy et al. [18] propose in the context of shape-from-shading has the same difficulty with image compositing (Fig. 6).

*Edge-preserving Filtering.* Our method is also related to Farbman’s edge-preserving filter [27] that minimizes an attachment term plus  $\int W_P(I_0) \|\nabla I\|^2$  where  $I_0$  is the input image. Farbman projects the formula on the  $x$  and  $y$  axes but we believe that it does not have a major impact on the results. More importantly, Farbman’s method and ours share the idea of using a modulation  $W_P$  that depends on fixed quantities and preserves the least-squares nature of the problem; Farbman uses the input image  $I_0$  and we use the target gradient field  $\mathbf{v}$ . Finally, our work has common points with Perona and Malik’s nonlinear anisotropic diffusion filter [28]:  $\partial I / \partial t = \operatorname{div}(W_P(\nabla I) \nabla I)$ . The difference is that our modulation term  $W_P$  is not a function of the image  $I$  which makes our equation linear, and we have a term  $\nabla I - \mathbf{v}$  instead of  $\nabla I$ , which can be interpreted as Perona and Malik “diffuse gradients” whereas we “diffuse integration residuals.”

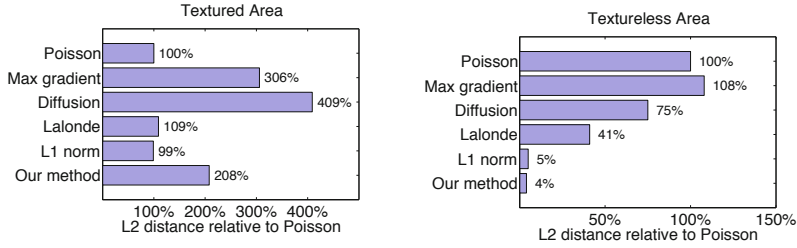


**Fig. 6.** We compare several approaches on an example where we composite a tree on a sky background. To test the robustness against selection inaccuracies, we introduce three errors (a): a small error on the left, a large error on the right, and the trunk is inserted in the ground. A direct copy-and-paste produces an image with visible seams in the sky region (b). Poisson compositing [11] (c), maximum gradient [1] (d), diffusion [14] (e), Photo Clip Art [16] (f), and robust Poisson reconstruction using the  $L_1$  norm [18] (g) generate seamless boundaries but suffer from bleeding artifacts where the selection cuts through the foliage and also at the contact between the trunk and the ground. In comparison, our method (h) produces artifact-free results. We provide more comparisons in supplemental material.

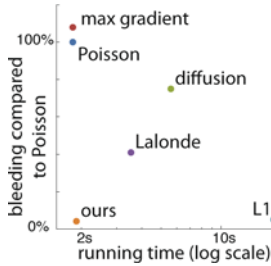
## 4 Results

We demonstrate our approach on a typical hand-made compositing scenario which may generate inaccurate selections (Fig. 6). We also show that our approach applies to heterogeneous panorama stitching [7] (Fig. 10) and image reshuffling [11] (Fig. 11). More results are in our supplemental material. All the results are computed using the same parameters unless otherwise specified. These settings performed well in all of our experiments. Parameter variations are also shown in the supplemental material.

*Quantitative Evaluation.* We use direct compositing and Poisson compositing as baselines to estimate how much bleeding occurs. For direct compositing, we directly copy pixel values and the result  $I_d$  exhibits visible seams but not bleeding. For Poisson compositing, we copy gradient values and solve the Poisson equation. The result  $I_P$  is seamless but colors leak where the selection is inaccurate. Then we consider an image  $I$ , pick a pixel  $\mathbf{p}$  in the potential leaking area, and compute:  $\|I(\mathbf{p}) - I_d(\mathbf{p})\| / \|I_P(\mathbf{p}) - I_d(\mathbf{p})\|$ . Expressed in percentages, 0% indicates no bleeding at all and 100% indicates as much bleeding as Poisson compositing. Figure 7 compares the results for several methods.



**Fig. 7.** We numerically evaluate bleeding introduced by different methods. We selected two  $11 \times 11$  regions in the tree example (Fig. 6), one within the textured area below the trunk and one in the sky on right of the foliage. We compute the  $L_2$  RGB difference between the image before and after compositing, normalized relative to Poisson compositing; that is, 100% indicates as much bleeding as Poisson and 0% indicates no bleeding. In the textured region (left), all methods bleed but the bleeding is masked by the high frequency texture. In the textureless area (right), most methods cause visible bleeding, which is particularly visible in this smooth region. The  $L_1$ -norm and our method achieve similarly low values which confirm minimal bleeding. But in a number of cases, the  $L_1$ -norm method introduces an undesirable color cast shown in the tree example, whereas our method yields a satisfying output.



**Fig. 8.** This plot locates each method according to its speed and how much bleeding it introduces in the sky region on Figure 6 as reported in Figure 7. Our method is as fast as the standard Poisson solver while introducing almost no bleeding. In comparison, the other methods are slower and generate color leaks. Note that the  $L_1$  method does not produce bleeding artifact on this example but it creates a severe color cast (Fig. 6).

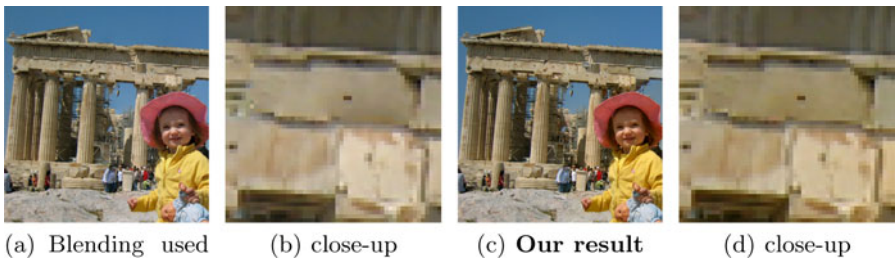
*Complexity.* We compute the final result in two linear steps. This is equivalent to a single linear system because  $I$  is a linear function of  $\mathbf{v}$  (Eq. 8) and  $\mathbf{v}$  is a linear function of  $B$  and  $F$  (Eq. 4). Further, only sparse operators are involved: divergence, curl, and weights that correspond to diagonal matrices. Compared to the Poisson equation, we solve for the same number of unknowns, that is, the number of pixels in  $I$ . The only overhead is the computation of  $\mathbf{v}$ , for which the number of unknowns is orders of magnitude smaller, since only pixels on the boundary are concerned. To summarize, our method has the same complexity as the Poisson equation. In comparison, nonlinear methods [18] require more complex iterative solvers. Figure 8 shows that our implementation achieves timings similar to the Poisson reconstruction, resulting in a run-time faster than most other implementations while introducing almost no bleeding.



**Fig. 9.** In some cases, when compared to the input (a), Poisson compositing (b) and our approach (c) discolor the pasted region. See text for details.



**Fig. 10.** For heterogeneous panorama, Photoshop Auto Blend [6] produces strong bleeding near the cut. In comparison, our method significantly improves the result. Our approach also performs better than to other methods on this challenging case as shown in supplemental material.



**Fig. 11.** Compared to the blending approach proposed by Cho et al. [11] (a,b), our approach (c,d) improves the result of image reshuffling. We used the same patch locations and boundaries as Cho et al. but applied our method which yields better results than the Poisson-based blending proposed in the original article [11]. In particular, our result produces more faithful colors but does have local color leaks as can be seen on the close-up (zoom of a region above the girl's hat). This result may be better seen in the supplemental material. Data courtesy of Tim Cho.

*Discussion.* Although our method produces high quality outputs, a close examination reveals that the boundary can be sometimes overly sharp. This minor issue is difficult to spot at first and less conspicuous than color leaks. Nonetheless, matching the sharpness of other edges in the image would be an interesting extension to this work. As other gradient-domain methods, our method can yield some discoloration (Fig. 9 and supplemental material). This effect is often desirable to achieve seamless blending. If one wishes to preserve the original colors, matting can be solution but it often requires a more careful user input. We also found that our approach is useful in challenging applications such as heterogeneous panorama stitching [7] where mismatches are common place (Fig. 10). In this case, we found that our method performs better with a smoother transition from seamless and leak-free compositing, which is achieved by setting  $\tau = 0.01$  in Equation (6).

## 5 Conclusion

We have described an image-compositing method that is robust to selection inaccuracies. The combination of low-curl boundaries and a weighted reconstruction based on visual masking produces artifact-free results on a broad range of inputs, in particular where other methods have difficulties. In addition, the solution is linear and has similar complexity to the standard Poisson equation. With robust results and speed, our method is a suitable replacement for the standard Poisson equation in many computer vision applications.

*Acknowledgments.* The authors thank Todor Georgiev for the link with the Poisson equation, Kavita Bala and George Drettakis for their discussion about visual masking, Aseem Agarwala and Bill Freeman for their help with the paper, Tim Cho and Biliانا Kaneva for helping with the validation, Medhat H. Ibrahim for the image of the Egyptian pyramids, Adobe Systems, Inc. for supporting Micah K. Johnson's research, and Ravi Ramamoorthi for supporting Michael Tao's work. This material is based upon work supported by the National Science Foundation under Grant No. 0739255 and No. 0924968.

## References

1. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. *ACM Trans. on Graphics* 22 (2003)
2. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D.H., Cohen, M.F.: Interactive digital photomontage. *ACM Trans. on Graphics* 23 (2004)
3. Georgiev, T.: Covariant derivatives and vision. In: *European Conf. on Computer Vision* (2006)
4. Jia, J., Sun, J., Tang, C.K., Shum, H.Y.: Drag-and-drop pasting. *ACM Trans. on Graphics* 25 (2006)
5. Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless image stitching in the gradient domain. In: *European Conf. on Computer Vision* (2006)
6. Agarwala, A.: Efficient gradient-domain compositing using quadtrees. *ACM Trans. on Graphics* 26 (2007)



7. Sivic, J., Kaneva, B., Torralba, A., Avidan, S., Freeman, W.T.: Creating and exploring a large photorealistic virtual space. In: IEEE Workshop on Internet Vision (2008)
8. Whyte, O., Sivic, J., Zisserman, A.: Get out of my picture! Internet-based inpainting. In: British Machine Vision Conf. (2009)
9. Finlayson, G.D., Drew, M.S., Lu, C.: Entropy minimization for shadow removal. *International Journal of Computer Vision* (2009)
10. Hays, J., Efros, A.A.: Scene completion using millions of photographs. *ACM Trans. on Graphics* 26 (2007)
11. Cho, T.S., Avidan, S., Freeman, W.T.: The patch transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2010)
12. Tappen, M.F., Adelson, E.H., Freeman, W.T.: Recovering intrinsic images from a single image. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27 (2005)
13. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.S.: On the removal of shadows from images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28 (2006)
14. Agrawal, A., Raskar, R., Chellappa, R.: What is the range of surface reconstructions from a gradient field? In: European Conf. on Computer Vision (2006)
15. Bhat, P., Zitnick, C.L., Cohen, M., Curless, B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. on Graphics* 28 (2009)
16. Lalonde, J.F., Hoiem, D., Efros, A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. *ACM Trans. on Graphics* 26 (2007)
17. Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., Fattal, R., Lischinski, D.: Coordinates for instant image cloning. *ACM Trans. on Graphics* 28 (2009)
18. Reddy, D., Agrawal, A., Chellappa, R.: Enforcing integrability by error correction using L-1 minimization. In: Computer Vision and Pattern Recognition (2009)
19. Drettakis, G., Bonneel, N., Dachsbacher, C., Lefebvre, S., Schwarz, M., Viaud-Delmon, I.: An interactive perceptual rendering pipeline using contrast and spatial masking. *Rendering Techniques* (2007)
20. Ramanarayanan, G., Ferwerda, J., Walter, B., Bala, K.: Visual equivalence: Towards a new standard for image fidelity. *ACM Trans. on Graphics* 26 (2007)
21. Vangorp, P., Laurijssen, J., Dutré, P.: The influence of shape on the perception of material reflectance. *ACM Trans. on Graphics* 26 (2007)
22. Ramanarayanan, G., Bala, K., Ferwerda, J.: Perception of complex aggregates. *ACM Trans. on Graphics* 27 (2008)
23. Su, S., Durand, F., Agrawala, M.: De-emphasis of distracting image regions using texture power maps. In: ICCV Workshop on Texture Analysis and Synthesis (2005)
24. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. *ACM Trans. on Graphics* 25 (2006)
25. Aubert, G., Kornprobst, P.: Mathematical problems in image processing: Partial Differential Equations and the Calculus of Variations. *Applied Mathematical Sciences*, vol. 147. Springer, Heidelberg (2002)
26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Trans. on Image Processing* 13 (2004)
27. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. on Graphics* 27 (2008)
28. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis Machine Intelligence* 12 (1990)

# Blind Reflectometry

Fabiano Romeiro and Todd Zickler

Harvard University  
33 Oxford St., Cambridge, MA, USA, 02138  
{romeiro,zickler}@seas.harvard.edu

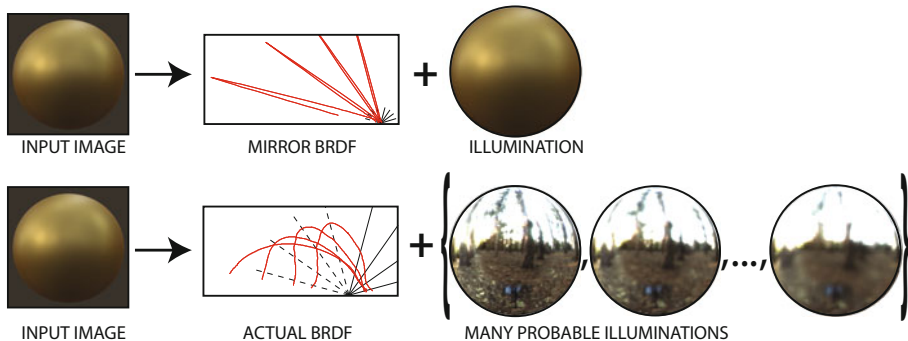
**Abstract.** We address the problem of inferring homogeneous reflectance (BRDF) from a single image of a known shape in an unknown real-world lighting environment. With appropriate representations of lighting and reflectance, the image provides bilinear constraints on the two signals, and our task is to blindly isolate the latter. We achieve this by leveraging the statistics of real-world illumination and estimating the reflectance that is most likely under a distribution of probable illumination environments. Experimental results with a variety of real and synthetic images suggest that useable reflectance information can be inferred in many cases, and that these estimates are stable under changes in lighting.

## 1 Introduction

The optical properties of a material often provide a clue for how it will behave when acted upon. They help inform us, for example, if the material is hard, soft, hot, cold, rigid, pliable, brittle, heavy, or lightweight. It makes sense, then, that people can infer materials' optical properties from their images; and building similar functionality into computer vision systems seems worthwhile.

The optical properties of many materials are adequately summarized by the bidirectional reflectance distribution function (BRDF), which describes how flux at a surface patch is absorbed and reflected over the output hemisphere. The BRDF provides a complete description of lightness, gloss, sheen, and so on; and in this paper, we explore when and how it can be recovered from an image. This task is made difficult by the fact that reflectance is confounded with shape, lighting, and viewpoint, all of which may be unknown. Even when the shape and relative viewpoint are provided (say, by contours, shadows, or other cues), the blind separation of BRDF from lighting is something that computer vision systems cannot yet do well.

This paper considers the following problem, depicted in Fig. 1. We are given a single high dynamic range (HDR) image of a known shape under an unknown, real-world lighting environment, and our task is to infer the material's BRDF. Our approach is to compute the BRDF that is most likely under a distribution of probable lighting environments—a strategy that is motivated by previous successes for other ill-posed vision problems, including color constancy and blind image deblurring. In our case, we show that by choosing an appropriate BRDF representation, we can leverage the statistics of real-world lighting to accurately infer materials' optical properties in a variety of lighting environments.



**Fig. 1.** Our goal is estimating the BRDF from an image of a known shape in unknown real-world lighting. *Top:* The trivial solution is a mirror-like BRDF, which exactly predicts the input for a carefully-crafted “blurry” environment. *Bottom:* To avoid this, we choose a BRDF that predicts the input for a distribution of probable environments.

## 2 Background and Related Work

People are quite adept at inferring reflectance information from images, and there have been a number of psychophysical studies that explore the underlying mechanisms [12, 7, 33, 31, 35]. Results suggest that people do not require contextual knowledge of the environment to infer reflectance [7], but that performance decreases when the directional statistics of the environment deviate significantly from those found in nature [7, 4, 5]. These findings provide motivation for our work, which also leverages the directional statistics of natural environments.

When it comes to computational approaches for recovering reflectance from images, most have been developed for controlled or known lighting (e.g., [30, 14, 9]). Fewer methods have been designed for cases where the lighting is not known, and of these, most assume reflectance to be well-represented by a pre-chosen “parametric” BRDF model, such as the Phong, Ward, or Lafortune models (e.g., [21, 36, 11]). Parametric BRDF models place considerable restrictions on reflectance, and as a result, they allow inferring quite a bit about a scene. For example, the method of Georghiadis [8] can simultaneously infer everything—shape, lighting and reflectance—provided that the material is well-represented by a simplified Torrance-Sparrow model (and that lighting consists of a moving point light source). While parametric models continue to improve (e.g., [20]), their use typically has two significant limitations. First, it severely restricts the space of materials (see [19, 32]); and second, because these models are non-linear in their parameters, the required computation ends up being model-specific and cannot easily be transferred from one material class to another.

An attractive alternative to parametric BRDF models is using a linear combination of reflectance basis functions. This way, the representation can be grown to include the entire world of BRDFs, at least in theory. Moreover, when object shape is known, it leads to a simple bilinear relationship between the unknown reflectance parameters (i.e., the coefficients in the basis) and the lighting

parameters. This bilinearity has already been exploited in both vision [13,10] and graphics (e.g., [25,18]), and it is the key to making our approach tractable.

The choice of bases for reflectance and lighting are important, and we discuss them in detail in subsequent sections. But once these choices are made, we obtain a bilinear inference problem that resembles others in vision: Given an image of a known shape, we must find probable lighting and reflectance parameters that could have created it. Color constancy and blind image deblurring can be formulated analogously [2,6,12], and our work leverages insight gained from their analyses. Specifically, instead of simultaneously estimating the BRDF and environment that best explain a given image, we obtain better results by estimating the BRDF that is most likely under a *distribution* of lighting environments (Fig. 1). This process is termed “MAP<sub>k</sub> estimation” in the context of blind deblurring [12], and the same basic idea forms the core of our approach.

A natural comparison for our approach is the framework of Ramamoorthi and Hanrahan [24,26], which uses spherical harmonics to represent lighting and reflectance and expresses their interaction as a convolution. Since spherical harmonics are eigenfunctions of the convolution operator, this leads to elegant closed-form expressions for the lighting and reflectance coefficients. But this representation cannot easily incorporate a meaningful prior probability distribution for natural lighting environments (see [5]), and it either requires that the entire 4D light field is available as input or that the BRDF can be restricted to being a “radially-symmetric” function over a one-dimensional domain.

Another natural comparison is the method of Haber et al. [10], which also represents lighting and reflectance using linear bases. Their approach differs in terms of its input and output (multiple images instead of one; spatially-varying BRDFs instead of uniform) and has two technical distinctions. It does not explicitly model the statistics of natural lighting, and it jointly estimates lighting and reflectance instead of marginalizing over a distribution of environments.

### 3 Approach

We assume all sources and reflecting surfaces in the environment to be far from the object in question so that the angular distribution of incident lighting does not vary over the object’s surface. This allows the unknown lighting  $L$  to be represented as an “environment map”—a positive-valued function on the sphere of directions;  $L: \mathbb{S}^2 \rightarrow \mathbb{R}^+$ . We also assume that the camera and object geometry are known, that mutual illumination is negligible, and that the unknown BRDF ( $F$ ) is isotropic. Then, a linear measurement made at pixel  $i$  can be written

$$I_i = \int_{\Omega} L_i(\omega) V_i(\omega) F_i(\omega) (n_i \cdot \omega) d\omega, \quad (1)$$

where  $n_i$  is the surface normal at the back-projection of pixel  $i$ ,  $L_i(\omega)$  is the hemisphere of the unknown lighting centered at direction  $n_i$ , and  $V_i(\omega)$  is a binary-valued hemispherical “visibility” function that encodes the object’s self-shadowing at the back-projection of  $i$  (e.g., [18,10]). Finally,  $F_i$  is a 2D slice of the unknown BRDF determined by the normal  $n_i$  and the local view direction.

Since everything in Eq. 1 is known except the lighting and BRDF, an image  $I = \{I_i\}$  imposes a set of constraints upon them. One approach to estimating the BRDF, then, is to define prior probability distributions for the unknown lighting  $p(L)$  and BRDF  $p(F)$  and find the functions that maximize the posterior

$$p(L, F|I) \propto p(I|L, F)p(L)p(F), \quad (2)$$

using a likelihood  $p(I|L, F)$  based on Eq. 1. This is closely related to the approach of Haber et al. [10], and it suffers from a preference for the trivial mirror-like solution. Any image can be perfectly explained by a mirror-like BRDF and a carefully crafted “blurry” environment that exactly matches the image [7], so the likelihood (and usually the posterior) are maximal for these functions.

In this paper we avoid this problem in the following manner. Instead of selecting the single BRDF/lighting pair that best explain an input image, we select the BRDF that is most likely under a *distribution* of lighting environments. We do this by computing the mean of the marginalized posterior:

$$F_{opt} \triangleq \int F p(F|I) dF = \int F \left( \int p(F, L|I) dL \right) dF. \quad (3)$$

The intuition here—adapted directly from the related problem of blind image deblurring [6,12]—is that instead of selecting a BRDF that perfectly explains the image for a *single* lighting environment (the trivial solution), we select one that reasonably explains the image for many probable lighting environments.

Evaluating the expression on the right of Eq. 3 requires prohibitive computation, and to make it feasible, we employ a variational Bayesian technique. Following [16,17] we approximate the posterior using a separable function,

$$p(L, F|I) \approx q(L, F) = q(L)q(F), \quad (4)$$

with components having convenient parametric forms. Given an input image, we compute the parameters of this approximate posterior using fixed point iteration, and then we trivially approximate the solution (Eq. 3) as the mean of  $q(F)$ .

Pursuing this approach requires suitable representations for lighting and reflectance. In particular, we require each to be a linear combination of basis functions, and we require the prior probability distributions of their coefficients to be well-approximated by exponential forms. We describe our choices next.

### 3.1 Representing Illumination

We represent spherical lighting using a wavelet basis. As depicted in Fig. 2 and following [34], we do this by mapping the sphere to a plane with an octahedral map [23] and using a Haar wavelet basis in this plane. Notationally, we write  $L = \sum_{m=1}^M \ell_m \psi_m$  with  $\psi_m$  the basis functions and  $\ell_m$  the corresponding coefficients. This choice of basis is motivated by the fact that statistics of band-pass filter coefficients of real-world lighting display significant regularity. Much like real-world images, the distributions of these coefficients are highly kurtotic, having

heavy tails [4,5]. Our choice is also motivated by the apparent utility of the related image statistics for tasks like compression, denoising, and deblurring.

To develop prior distributions for the coefficients  $\ell \triangleq \{\ell_m\}$ , we collected 72 environments (nine from the ICT Graphics Lab[1] and the remainder from the SIBL Archive[2]), normalized each so that it integrates to one, and studied the coefficient distributions at different scales. Like Dror et al. [4,5], we found these statistics to be notably non-stationary, especially at coarser scales. Figure 2 shows empirical distributions and parametric fits for a variety of scales using a  $32 \times 32$  discretization of the sphere. At the finest scales (scales 4 and 5), the distributions are quite stationary, and we employ a single zero-mean Gaussian mixture for all of the coefficients at each scale (with 4 and 5 components, respectively). At the middle scales (scales 2 and 3), the statistics change significantly depending on elevation angle and basis type (vertical, diagonal, horizontal), and accordingly we use distinct distributions for each basis type both above and below the horizon. Each distribution is a zero-mean Gaussian mixture, and we use three components for groups in scale 3 and two components for groups in scale 2. Finally, at the coarsest scale (scale 1) we use zero-mean, two-component Gaussian mixtures for the diagonal and horizontal basis types, and a Gaussian rectified at a negative value for the vertical basis type to capture the fact that lighting is dominant from above. Note that the DC value  $\ell_1$  is the same in all cases since the illuminations are normalized. Additional details are in [28].

With these definitions we can write our illumination prior as

$$p(\ell) = \prod_{m=2}^M \sum_{n=1}^{N_m} \pi_{nm} p_{nm}(\ell_m), \quad (5)$$

with  $N_m$  the number of mixture components for coefficient  $m$  and  $\pi_{nm}$  the mixing weights. The group structure described above is implicit in this notation: All coefficients in any one group share the same  $N_m$ ,  $\pi_{nm}$  and  $p_{nm}$ .

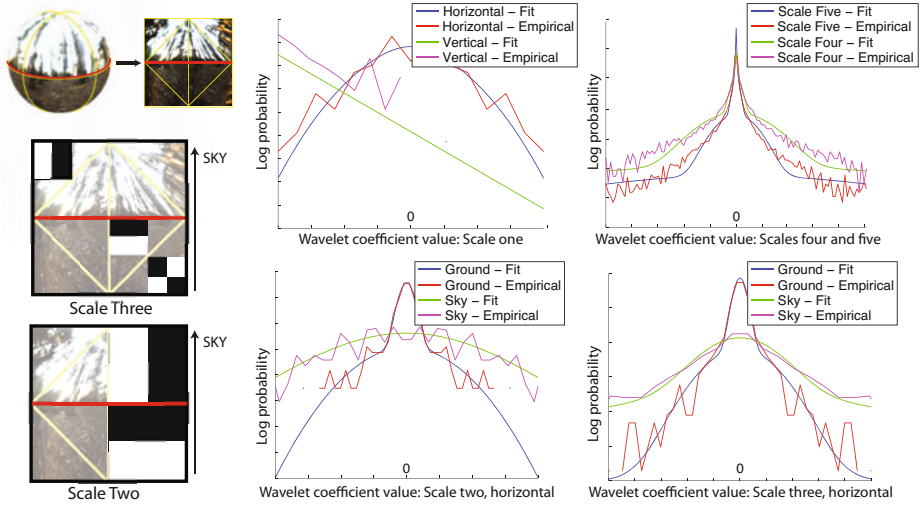
### 3.2 Representing Reflectance

We represent BRDFs as a linear combination of non-negative basis functions learned through non-negative matrix factorization (NMF) of all 100 materials in the MERL/MIT database [15]. This produces the linear representation  $F = \sum_{k=1}^K f_k \phi_k$  and has the advantage of allowing non-negativity constraints on the recovered BRDF  $F$  to be naturally enforced through non-negativity constraints on the coefficients  $f \triangleq \{f_k\}$ . Also, we find that the empirical distributions of the resulting coefficients  $f_k$  can be well-approximated by exponentials (see Fig. 3, right), making them well-suited for inference using variational Bayes.

Each BRDF in the database is represented using a  $90 \times 90 \times 180$  discretization of the 3D isotropic BRDF domain, parameterized in terms of the half-vector and difference-vector [29]. For computational convenience, we reduce each material to

<sup>1</sup> <http://www.debevec.org/probes>; <http://gl.ict.usc.edu/Data/HighResProbes>

<sup>2</sup> <http://www.hdrilabs.com/sibl/archive.html>

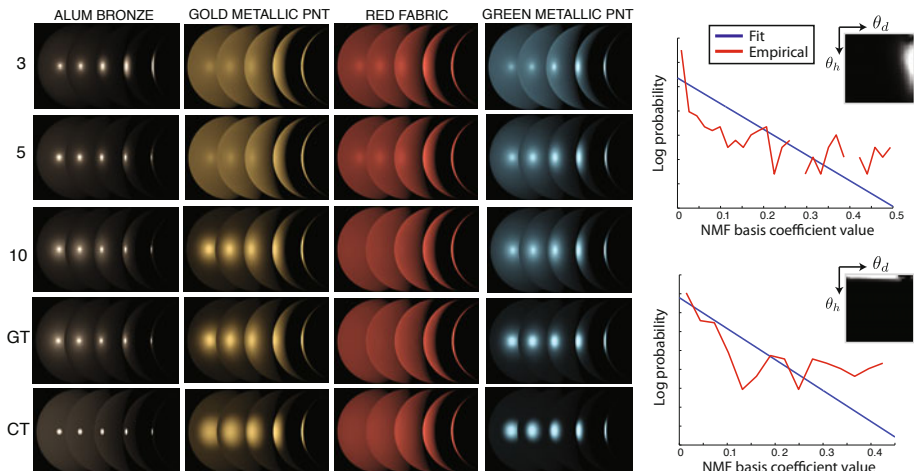


**Fig. 2.** *Left:* We represent lighting using a Haar wavelet basis on the octahedral domain [23] discretized to  $32 \times 32$ . Statistics of wavelet coefficients are non-stationary, so we fit distinct distributions for coefficients above and below the horizon. *Right:* Empirical distributions and their parametric fits for a variety of wavelet coefficient groups.

the  $90 \times 90$  bivariate domain of Romeiro et al. [27] and scale it to have a maximum value of one before computing the NMF. The bivariate reduction allows handling the entire database (and potentially much more) without resulting to out-of-core methods, and as shown in [27], it has a limited effect on accuracy. The resulting basis functions are defined on the two dimensional domain  $(\theta_h, \theta_d) \in [0, \pi/2) \times [0, \pi/2)$ , where  $\theta_h$  and  $\theta_d$  are the halfway angle and difference angle, respectively (see [29]). We can visualize the basis functions as images, and two of them are in the right of Fig. 3. In this visualization, specular reflection is at small halfway angles (top edge), grazing effects are at large difference angles (top-right corner) and retro-reflection is at small difference angles (left edge).

The left of Fig. 3 qualitatively evaluates the NMF model’s fit to the original BRDF data for different numbers ( $K$ ) of basis functions. We also compare to a parametric BRDF model (Cook-Torrance) as fit by Ngan et al. [19]. While it remains perceptually distinguishable from ground truth for some materials, we find that the NMF model’s fit with ten basis functions ( $K = 10$ ) provides a good balance between complexity and accuracy. It provides fits that are comparable to the Cook-Torrance model, but it is linear, which is important to our approach.

We can also evaluate the fit quantitatively by computing RMS error in the BRDF domain. According to this metric, the NMF approach significantly outperforms the parametric model. The mean and median RMS error computed using the green channels of all materials are 1.58 and 0.45 for the NMF model, and 46.91 and 17.11 for the Cook-Torrance fit. Some of this significant difference is due to the fact that we have chosen to perform NMF based on  $L_2$  cost



**Fig. 3.** *Left:* Qualitative evaluation of the NMF BRDF model. Top to bottom: NMF model with 3, 5, and 10 basis functions; ground truth; and Cook-Torrance fit from [19]. *Right:* Empirical distributions and parametric fits for NMF coefficients corresponding to basis elements that roughly account for grazing (top) and specular (bottom) effects.

in the BRDF domain, whereas the parametric Cook-Torrance fit is performed with an approximate perceptual metric [19]. In fact, one can view the metric used in NMF as a choice that can be tuned for each application. If one ultimately seeks to infer BRDFs for the purposes of material recognition, then the  $L_2$  cost used here may be preferred. If the inferred BRDF is to be used for image synthesis, however, it may be more desirable to use a perceptual metric (e.g., [35]) within a kernel-NMF framework (e.g., [3]).

Having computed basis functions  $\phi_k$  and the parameters  $\lambda_k$  of the coefficient distributions, we obtain the following prior distribution for reflectance:

$$p(f) = \prod_{k=1}^K \lambda_k \exp(-\lambda_k f_k), \quad f_k \geq 0. \quad (6)$$

### 3.3 A Bilinear Likelihood

Having defined linear representations of the lighting and reflectance, we can write an expression for the likelihood of their coefficients given a particular image. We begin by updating the imaging model to include a camera exposure parameter ( $\gamma$ ) and a crude model for noise:

$$I_i = \gamma \int_{\Omega} L_i(\omega) V_i(\omega) F_i(\omega) (n_i \cdot \omega) d\omega + \epsilon, \quad (7)$$

with  $\epsilon \sim N(0, \sigma^2)$ . The exposure parameter compensates for the difference between the absolute scale of the intensity measurements and the combined scale of



the illumination and reflectance functions. This is important because the prior distributions for lighting and reflectance are estimated from normalized data while the intensity measurements may be at an arbitrary scale.

Substituting  $L = \sum \ell_m \psi_m$  and  $F = \sum f_k \phi_k$  into this expression, one can re-write this as (see details in [28]):

$$I_i = \gamma \ell^T M_i f + \epsilon, \quad (8)$$

where the per-pixel matrices  $M_i$  are determined by the shape  $(V_i, n_i)$ , view direction, and the lighting and reflectance basis functions  $\{\psi_m\}$  and  $\{\phi_k\}$ . For an input image of a known shape, these matrices can be pre-computed, and we assume them to be constant and known.

By treating the pixels of an input image as independent samples, this measurement model leads directly to our desired expression for the likelihood of a set of model parameters given image  $I$ :

$$p(I|\ell, f, \sigma, \gamma) = \prod_{i=1}^N \frac{\sigma^{-1}}{\sqrt{2\pi}} \exp\left(-\frac{\sigma^{-2}}{2}(I_i - \gamma \ell^T M_i f)^2\right). \quad (9)$$

We treat the exposure and noise variance  $(\gamma, \sigma^2)$  as model parameters to be estimated along with illumination and reflectance, and for these we define prior distributions  $p(\sigma^{-2}) \sim \Gamma(a, b)$  and  $p(\gamma) \sim \text{Exp}(\lambda_\gamma)$ .

### 3.4 Inference

The definitions of the previous sections (Eqs. [5], [6], [9], and the noise and exposure priors) provide everything we need to write the posterior

$$p(\ell, f, \gamma, \sigma^{-2}|I) \propto p(I|\ell, f, \gamma, \sigma^{-2})p(\ell)p(f)p(\gamma)p(\sigma^{-2}). \quad (10)$$

As described in Sect. [3], we wish to marginalize over lighting (as well as noise, and exposure) and compute the mean of the marginalized posterior. Following Miskin and MacKay [16], [17], we do this by approximating the posterior with a separable function,  $p(\theta|I) \approx q(\theta) = q(\ell)q(f)q(\sigma^{-2})q(\gamma)$ , with  $\theta \triangleq (\ell, f, \sigma^{-2}, \gamma)$ . The function  $q(\theta)$  is computed by minimizing a cost based on the Kullback-Leibler divergence between it and the posterior [16], [17]:

$$\int q(\theta) \left( \log \frac{q(\ell)}{p(\ell)} + \log \frac{q(f)}{p(f)} + \log \frac{q(\sigma^{-2})}{p(\sigma^{-2})} + \log \frac{q(\gamma)}{p(\gamma)} - \log p(I|\theta) \right) d\theta \quad (11)$$

We provide an overview of the optimization here, and details can be found in [28]. The basic idea is to use coordinate descent, with each distribution  $q(\cdot)$  being updated using the current estimates of the others. The update equations are derived by integrating all of the terms but one in Eq. [11] (the one containing  $q(f)$ , say), taking the derivative with respect to the remaining distribution ( $q(f)$

in this example) and equating the result to zero. In our case, this procedure reveals that the approximating distributions  $q(\cdot)$  are of the following forms

$$q(f) = \prod q_k(f_k), \text{ with } q_k \sim N_R(u_k, w_k), \quad (12)$$

$$q(\ell) = \prod q_m(\ell_m), \text{ with } q_m \sim \begin{cases} N(u_m, w_m) & \text{if } m \neq 3 \\ N_{RC}(u_m, w_m, T) & \text{otherwise,} \end{cases} \quad (13)$$

$$q(\gamma) \sim N_R(u_\gamma; w_\gamma) \text{ and } q(\sigma^{-2}) \sim \Gamma(\sigma^{-2}; a_p, b_p), \quad (14)$$

where  $N_R$  is a Gaussian distribution rectified at 0, and  $N_{RC}$  is a Gaussian distribution rectified at  $T$ . The same procedure also provides closed form expressions for the updated parameters of each distribution  $q(\cdot)$  in terms of the current parameters of the others (see [28]). One strategy, then, is to cycle through these distributions, updating each in turn; but as described in [16,17], convergence can be accelerated by updating all parameters in parallel and then performing a line search between the current and updated parameter-sets.

Specifically, we define intermediate variables that are sufficient to determine all of the distribution parameters:  $\Phi = (\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_7)$  where  $\Phi_1$  and  $\Phi_2$  are  $K$ -vectors such that  $\Phi_1(k) = \frac{u_k}{w_k}$  and  $\Phi_2(k) = \log(\frac{1}{w_k})$ ;  $\Phi_3$  and  $\Phi_4$  are  $M$ -vectors such that  $\Phi_3(m) = \frac{u_m}{w_m}$  and  $\Phi_4(m) = \log(\frac{1}{w_m})$ ;  $\Phi_5 = \log(\frac{b_p}{a_p})$ ;  $\Phi_6 = \frac{u_\gamma}{w_\gamma}$ ; and  $\Phi_7 = \log(\frac{1}{w_\gamma})$ . These intermediate variables are iteratively updated according to Algorithm 1, and once they converge, they determine the distribution  $q(f)$ , whose mean is the BRDF we seek. The noise variable  $\Phi_5$  is not updated at every iteration, but only when the other variables have converged at the current noise level. This is a strategy borrowed from Miskin’s implementation [3].

We initialize the algorithm with the posterior means  $\{u_k^{(0)}\}$  and  $\{u_m^{(0)}\}$  corresponding to a random BRDF and lighting environment, respectively. The initial posterior variances  $\{w_m^{(0)}\}$  and  $\{w_k^{(0)}\}$  are set to relatively large values ( $10^{-1}$ ) to account for the uncertainty in our initial estimates. Exposure parameters  $u_\gamma$  and  $w_\gamma$  are initialized to 1 and 10 respectively. Finally, parameter  $\frac{b_p}{a_p}$  is initialized to 1 so that we have a broad initial posterior on the inverse noise variance.

## 4 Evaluation and Results

We begin our evaluation using images synthesized [4] with the MERL/MIT BRDF data and our collection of measured illumination environments. Using these tools, we can render HDR images for input to our algorithm as well as images with the recovered BRDFs for comparison to ground truth.

There is a scale ambiguity for each image because we can always increase the overall brightness of the illumination by making a corresponding decrease in the BRDF. Accordingly, we only seek to estimate the BRDF up to scale. We

<sup>3</sup> [http://www.inference.phy.cam.ac.uk/jwm1003/train\\_ensemble.tar.gz](http://www.inference.phy.cam.ac.uk/jwm1003/train_ensemble.tar.gz)

<sup>4</sup> PBRT: <http://www.pbrt.org/>

---

**Algorithm 1.** Fit ensemble of approximating distributions

---


$$\phi_1^{(0)}(k) \leftarrow \frac{u_k^{(0)}}{w_k^{(0)}}, \phi_2^{(0)} \leftarrow \log\left(\frac{1}{w_k^{(0)}}\right), \phi_3^{(0)}(m) \leftarrow \frac{u_m^{(0)}}{w_m^{(0)}}, \phi_4^{(0)} \leftarrow \log\left(\frac{1}{w_m^{(0)}}\right)$$

$$\phi_5^{(0)} \leftarrow \log(1), \phi_6^{(0)} \leftarrow \frac{u_\gamma^{(0)}}{w_\gamma^{(0)}}, \phi_7^{(0)} \leftarrow \log\left(\frac{1}{w_\gamma^{(0)}}\right), i = 0$$

**repeat**

**repeat**

$\Phi^* = \text{Update}(\Phi^{(i)})$  (see [28] for update equations),  $\Delta\Phi = \Phi^* - \Phi^{(i)}$

$\alpha^* = \arg \min_{\alpha} C_{KL}(\Phi^{(i)} + \alpha\Delta\Phi)$  (see [28] for cost expression  $C_{KL}$ )

$\Phi^{(i+1)} = \Phi^{(i)} + \alpha^* \Delta\Phi$ ,  $\Phi_5^{(i+1)} = \Phi_5^{(i)}$ ,  $i = i + 1$

**until**  $|C_{KL}^{(i+1)} - C_{KL}^{(i)}| < 10^{-4}$

$\Phi_5^{(i)} = \Phi_5^{(i-1)} + \alpha^* \Delta\Phi_5$

**until**  $\|\Phi_5^{(i)} - \Phi_5^{(i-1)}\| < 10^{-4}$

---

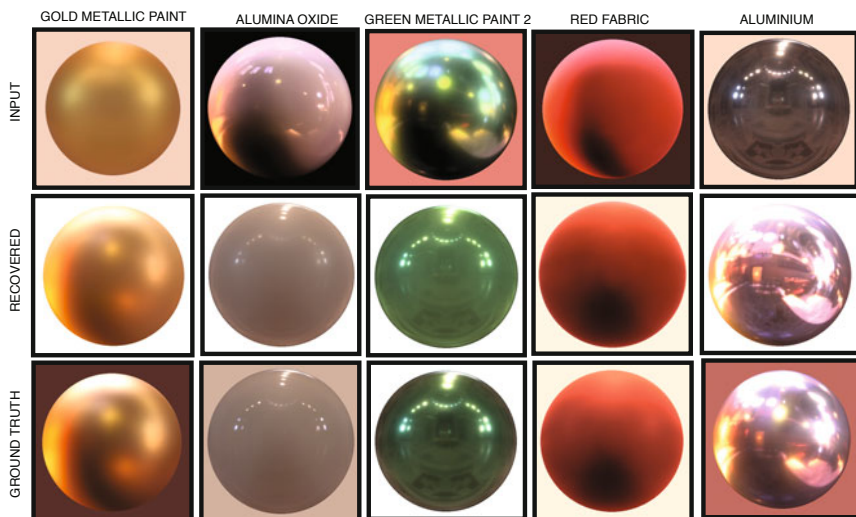
also ignore wavelength-dependent (color) effects by performing inference on the luminance channel and recovering a monochrome BRDF as output. Inferring wavelength-dependent reflectance effects (i.e., a spectral BRDF) would require solving the color constancy problem in conjunction with the reflectometry problem, and we leave this problem for future research.

While we operate in grayscale, we display the input and output using color in this paper. The displayed input is the color image prior to extracting luminance, and the displayed output is the outer product of the recovered monochromatic BRDF and the RGB vector that provides the best fit to the ground truth. This visualization strategy produces artifacts in some cases. For example, the color-visualization of the recovered red material in Fig. 6 does not (and cannot) match the highlight colors of the reference image.

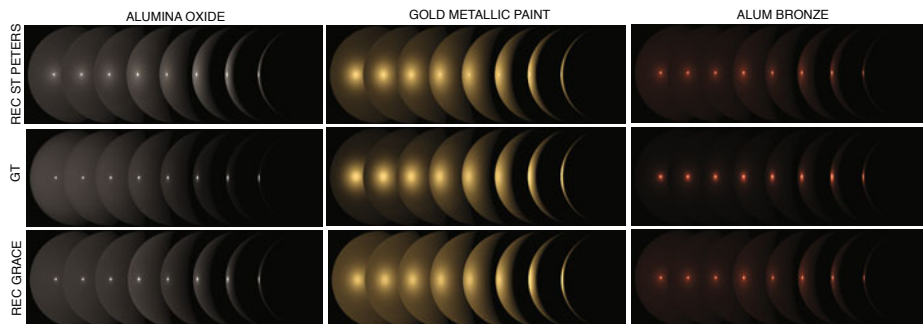
Given a rendered input image of a defined shape (we use a sphere for simplicity), we collect observations from 12,000 normals uniformly sampled on the visible hemisphere. We discard normals that are at an angle of more than 80 degrees from the viewing direction (since the signal to noise ratio is very low at these points) as well as normals that are close to the poles of our parametrization of the sphere (as Eq. 8 is not a good approximation in these regions). This results in an observation vector  $I$  of length 8,600.

Each column of Fig. 4 shows a BRDF recovered from a single input image synthesized with either the St. Peter’s Basilica or Grace Cathedral environment. Following [27], the recovered BRDFs are compared to ground truth by synthesizing images in a novel environment, and close inspection shows them to be visually quite accurate. Figure 5 further explores stability under changes in lighting. For this, we run our algorithm twice for each material using two different environments and compare the recovered BRDFs. We visualize these BRDFs along with ground truth by using them to synthesize a “spheres image” inspired by [32]. The BRDF estimates are quite consistent across the two environments, and they provide imperfect but reasonable approximations to ground truth.

The same procedure was applied to the captured data from [27]. As above we operate on the luminance channel of HDR images and estimate a monochrome

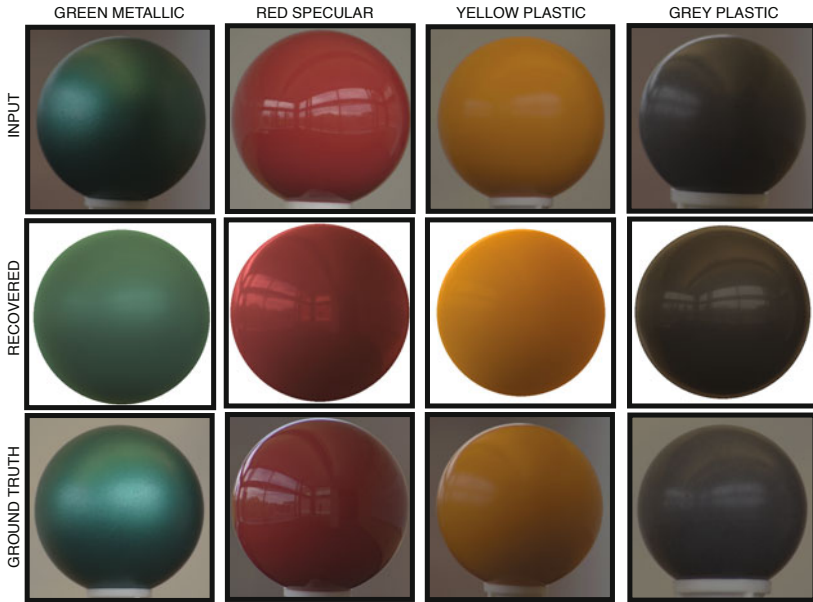


**Fig. 4.** Evaluation with synthetic input. *Top*: Single image used as input. *Middle*: Appearance predicted in a novel environment using the recovered BRDF. *Bottom*: Ground truth image in the same novel environment.



**Fig. 5.** Stability under changes in lighting: BRDFs recovered when the same material is seen in different environments. *Top to bottom*: BRDF recovered in the St. Peter’s environment; ground truth; and BRDF recovered in the Grace Cathedral environment.

BRDF, but now we visualize the output in color by taking the outer product of the monochrome BRDF and the median RGB color of the input image. Figure 6 shows results with the BRDFs recovered from single input images (top row) being used to render synthetic images of the same material under novel environments (more precisely, the same environment from a different viewpoint). Accuracy is assessed by comparing these synthetic images to real images captured in the same novel environments. While the recovered reflectance is clearly distinguishable from ground truth, we see that useful qualitative reflectance information is still



**Fig. 6.** Evaluation with captured input. *Top*: Image used as input. *Middle*: Appearance predicted in a novel environment using the recovered BRDF. *Bottom*: Ground truth images captured in the same novel environments. (Image data provided by [27].)

obtained. Based on the inferred BRDFs, for example, it would be straightforward to create an ordering of the four materials based on gloss.

These results reveal two limitations of the approach. First, one should expect less accuracy when the input image contains significant mesostructure (e.g., **green metallic**) or texture because these small variations effectively increase noise. Second, performance will be diminished when the illumination is “inadequate”, meaning that it does not induce significant specular, grazing, and/or retro reflections, and does not sufficiently constraint the BRDF (e.g., **red specular** and **yellow plastic**). This latter limitation is consistent with perceptual findings [7] and frequency-domain arguments [26], and it has been documented for cases in which the environment is known [27]. Romeiro et al. [27] also describe why quantitative analysis of the conditions for adequate illumination are difficult: Unlike the spherical harmonic approach [26], lighting and reflectance are not related by a convolution operator in the present case. Perhaps a quantitative description of the conditions for “adequate illumination” in material recognition will be a fruitful direction for future work.

## 5 Discussion

Our results suggest that for a range of homogeneous diffuse and glossy materials, reflectance information can be inferred from unknown real-world illumination when the object shape is known. They also suggest that these estimates are fairly stable when the illumination undergoes a change.

The approach has at least two features worth highlighting. First, it uses a linear basis for reflectance. This allows a seamless trade between complexity and accuracy, and it is very different from “parametric” BRDF models (Phong, etc.) that are non-linear in their variables and are only suitable for a particular material class. Second, it is an inference system built upon a probabilistic generative image model, and this makes it amenable to combination with other contextual cues and vision subsystems. In particular, we might explore combinations with shape-from-X techniques (shading, contours, shadows, etc.) to assess how well reflectance can be recovered when shape is not known *a priori*.

**Acknowledgments.** The authors thank Bill Freeman and Yair Weiss for helpful discussions regarding bilinear inference problems. This work was supported by the Office of Naval Research through award N000140911022, NSF Career Award IIS-0546408, and a fellowship from the Alfred P. Sloan Foundation.

## References

1. Beck, J., Prazdny, S.: Highlights and the perception of glossiness. *Perception & Psychophysics* 30(4) (1981)
2. Brainard, D., Freeman, W.: Bayesian color constancy. *J. Opt. Soc. Am. A* 14(7) (1997)
3. Ding, C., Li, T., Jordan, M.: Convex and semi-nonnegative matrix factorizations. *IEEE T. Pattern Anal.* 32(1) (2010)
4. Dror, R.: Surface reflectance recognition and real-world illumination statistics. Ph.D. thesis, Massachusetts Institute of Technology (2002)
5. Dror, R., Willsky, A., Adelson, E.: Statistical characterization of real-world illumination. *J. Vision* 4 (2004)
6. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM T. Graphics (Proc. ACM SIGGRAPH)* 25(3) (2006)
7. Fleming, R., Dror, R.O., Adelson, E.H.: Real-world illumination and the perception of surface reflectance properties. *J. Vision* 3(5) (2003)
8. Georgiades, A.: Recovering 3-D shape and reflectance from a small number of photographs. In: *Proc. Eurographics Workshop on Rendering* (2003)
9. Ghosh, A., Achutha, S., Heidrich, W., O’Toole, M.: BRDF acquisition with basis illumination. In: *Proc. IEEE Int. Conf. Computer Vision* (2007)
10. Haber, T., Fuchs, C., Bekaer, P., Seidel, H., Goesele, M., Lensch, H.: Relighting objects from image collections. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2009)
11. Hara, K., Nishino, K., Ikeuchi, K.: Mixture of spherical distributions for single-view relighting. *IEEE T. Pattern Anal.* 30(1) (2008)
12. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Understanding and evaluating blind deconvolution algorithms. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2009)
13. Mahajan, D., Ramamoorthi, R., Curless, B.: A theory of spherical harmonic identities for BRDF/lighting transfer and image consistency. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006. LNCS, vol. 3954*, pp. 41–55. Springer, Heidelberg (2006)

14. Marschner, S., Westin, S., Lafortune, E., Torrance, K., Greenberg, D.: Image-based BRDF measurement including human skin. In: Proc. Eurographics Symposium on Rendering (1999)
15. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. *ACM T. Graphics (Proc. ACM SIGGRAPH)* 22(3) (2003)
16. Miskin, J.: Ensemble Learning for Independent Component Analysis. Ph.D. thesis, University of Cambridge (2000)
17. Miskin, J., MacKay, D.: Ensemble learning for blind source separation. *Independent Component Analysis: Principles and Practice* (2001)
18. Ng, R., Ramamoorthi, R., Hanrahan, P.: Triple product wavelet integrals for all-frequency relighting. *ACM T. Graphics (Proc. ACM SIGGRAPH)* 23(3) (2004)
19. Ngan, A., Durand, F., Matusik, W.: Experimental analysis of BRDF models. In: Eurographics Symposium on Rendering (2005)
20. Nishino, K.: Directional statistics BRDF model. In: Proc. IEEE Int. Conf. Computer Vision (2009)
21. Nishino, K., Zhang, Z., Ikeuchi, K.: Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In: Proc. IEEE Int. Conf. Computer Vision (2001)
22. Pellacini, F., Ferwerda, J., Greenberg, D.: Toward a psychophysically-based light reflection model for image synthesis. In: Proc. ACM SIGGRAPH (2000)
23. Praun, E., Hoppe, H.: Spherical parametrization and remeshing. *ACM T. Graphics* 22(3) (2003)
24. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: *Proceedings of ACM SIGGRAPH*, pp. 117–128 (2001)
25. Ramamoorthi, R., Hanrahan, P.: Frequency space environment map rendering. *ACM Transactions on Graphics (TOG)* 21(3), 517–526 (2002)
26. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for reflection. *ACM T. Graphics* 23(4) (2004)
27. Romeiro, F., Vasilyev, Y., Zickler, T.: Passive reflectometry. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 859–872. Springer, Heidelberg (2008)
28. Romeiro, F., Zickler, T.: Ensemble learning for reflectometry. Tech. Rep. TR-06-10, Harvard School of Engineering and Applied Sciences (2010), <ftp://ftp.deas.harvard.edu/techreports/tr-06-10.pdf>
29. Rusinkiewicz, S.: A new change of variables for efficient BRDF representation. In: Eurographics Rendering Workshop '98 (1998)
30. Sato, Y., Wheeler, M., Ikeuchi, K.: Object shape and reflectance modeling from observation. In: *Proceedings of ACM SIGGRAPH* (1997)
31. Sharan, L., Li, Y., Motoyoshi, I., Nishida, S., Adelson, E.: Image statistics for surface reflectance perception. *J. Opt. Soc. Am. A* 25(4) (2008)
32. Stark, M., Arvo, J., Smits, B.: Barycentric parameterizations for isotropic BRDFs. *IEEE T. Vis. Computer Graphics* 11(2) (2005)
33. Todd, J., Farley Norman, J., Mingolla, E.: Lightness constancy in the presence of specular highlights. *Psychological Science* 15(1), 33–39 (2004)
34. Wang, R., Ng, R., Luebke, D., Humphreys, G.: Efficient wavelet rotation for environment map rendering. In: Eurographics Symposium on Rendering (2006)
35. Wills, J., Agarwal, S., Kriegman, D., Belongie, S.: Toward a perceptual space for gloss. *ACM T. Graphics* 28(4) (2009)
36. Yu, T., Wang, H., Ahuja, N., Chen, W.: Sparse lumigraph relighting by illumination and reflectance estimation from multi-view images. In: Eurographics Symposium on Rendering (2006)

# Photometric Stereo for Dynamic Surface Orientations

Hyeongwoo Kim<sup>1</sup>, Bennett Wilburn<sup>2</sup>, and Moshe Ben-Ezra<sup>2</sup>

<sup>1</sup> KAIST, Daejeon, Republic of Korea  
hyeongwoo.kim@kaist.ac.kr

<sup>2</sup> Microsoft Research Asia, Beijing, China  
{bennett.wilburn,mosheb}@microsoft.com

**Abstract.** We present a photometric stereo method for non-rigid objects of unknown and spatially varying materials. The prior art uses time-multiplexed illumination but assumes constant surface normals across several frames, fundamentally limiting the accuracy of the estimated normals. We explicitly account for time-varying surface orientations, and show that for unknown Lambertian materials, five images are sufficient to recover surface orientation in one frame. Our optimized system implementation exploits the physical properties of typical cameras and LEDs to reduce the required number of images to just three, and also facilitates frame-to-frame image alignment using standard optical flow methods, despite varying illumination. We demonstrate the system’s performance by computing surface orientations for several different moving, deforming objects.

## 1 Introduction

Photometric stereo [16] uses multiple images of an object illuminated from different directions to deduce a surface orientation at each pixel. In this work, we address accuracy limits for estimating surface orientations for dynamic scenes using photometric stereo, and in particular for deforming (non-rigid) objects. Photometric stereo for moving scenes is complicated because the world has only one illumination condition at a time, and the scene may move as one tries to change the lighting. Using a color camera and colored lights from different directions, one can measure shading for light from three directions in one image, but this only determines the surface orientation if the object reflectance is known and uniform.

The prior art for photometric stereo with deforming objects of varying or unknown materials uses time-multiplexed illumination (TMI) [15] to capture video while changing the lighting from frame to frame. Subsequent frames are aligned using optical flow, and the surface orientation is assumed to be constant across those frames. Assuming fixed surface normals for dynamic scenes is a contradiction and represents a fundamental accuracy limit for current TMI photometric stereo methods for non-rigid objects. For commonly occurring motions, we show this leads to significant errors in estimated surface orientations and albedos.



We present a photometric stereo method for deforming objects that is robust to changing surface orientations. We use time and color illumination multiplexing with three colors, but ensure an instantaneous measurement in every frame, either of the surface normal or of a subset of the material imaging properties. We use optical flow to account for varying motion at each pixel. Unlike the prior art, given accurate optical flow, our estimated surface normals are not corrupted if those normals are time-varying. Optical flow for TMI video is challenging because the intensity constancy assumption does not hold. Our optimized system implementation ensures constant illumination in one color channel, facilitating optical flow between subsequent frames using standard methods, despite varying illumination. Photometric stereo results for several deforming objects verify the performance of the system.

## 2 Background

Although the literature on shape capture of deforming objects is vast, Nehab et al. [9] observed that orientation-sensing technologies like photometric stereo are more accurate for high frequency shape details, while range sensing technologies (such as multi-view stereo) are better for low frequency shape. They devised an efficient method to combine the two forms of data to estimate precise geometry. These two forms of shape estimation are fundamentally different, so we will restrict our review to photometric stereo methods. The traditional photometric stereo [16] formulation assumes a static object imaged by a fixed camera under varying illumination directions. For a moving rigid object, many methods combine shading information with motion or multi-view stereo, assuming either fixed illumination (for example, [11,18]) or even varying lighting [6]. In this work, however, we aim to measure the surface orientation of deforming (non-rigid) objects, whose shape may vary from frame to frame, and whose motion cannot be represented simply as a rigid transformation.

Petrov [10] first addressed photometric stereo with multi-spectral illumination. One challenge of multi-spectral photometric stereo is the camera color measurements depend not only on the surface normal and light direction, but also on the interaction between the light spectra, material spectral responses, and the camera color spectral sensitivities. The method of Kontsevich et al. calibrates these dependencies using the image of the object itself, assuming the surface has a sufficient distribution of orientations [7]. The technique works for uncalibrated objects and materials, but is sensitive to the object geometry and unwieldy for multi-colored objects. Hernández et al. [5] presented a photometric stereo method that uses colored lights to measure surface normals on deforming objects. They show impressive results capturing time-varying clothing geometry, but the method requires the objects to consist of a single uniform material.

Wenger et al. [15] propose using time-multiplexed illumination (TMI) for photometric stereo. Their system uses high-speed video of an actor under 156 different lighting conditions and aligns images to target output frames using optical flow. Their goal is performance relighting, but they also compute surface normals and albedos for deforming objects and changing materials. Vlastic et al. [13]

extend this idea to multi-view photometric stereo, using an array of cameras and time-multiplexed basis lighting. Both methods assume fixed normals across the images used to compute each output frame. Weise et al. [14] explicitly handle deforming objects with TMI to sense depth (not orientation) using a stereo phase-shift structured light technique.

De Decker et al. [3] combine time and color multiplexing to capture more illumination conditions in fewer frames than TMI alone. Their photometric stereo method does not explicitly address changing surface orientations. It also neglects light–sensor crosstalk, causing significant errors for common cameras (including theirs). The method computes optical flow using a filter that “removes the lighting, but preserves the texture” by normalizing for local brightness and contrast. For photometric stereo, however, the image texture and lighting are not separable. Imagine the dimples on a golf ball lit from one side, and then the other—the changing texture is itself the shading information. Assuming it to be a fixed feature for optical flow will corrupt the estimated normals.

In this paper, we describe how to use time and color multiplexing for photometric stereo given changing surface orientations. We start by adding a changing surface normal to the traditional photometric stereo formulation.

### 3 Dynamic Photometric Stereo with Time and Color Multiplexed Illumination

The observed intensity of a Lambertian surface with surface normal  $\hat{\mathbf{n}}$ , illuminated from direction  $\hat{\mathbf{l}}$  is

$$I = \hat{\mathbf{l}} \cdot \hat{\mathbf{n}} \int S(\lambda)\rho(\lambda)\nu(\lambda)d\lambda, \quad (1)$$

where  $S(\lambda)$  is the light energy distribution versus wavelength,  $\rho(\lambda)$  is the material spectral reflectance, and  $\nu(\lambda)$  is the camera spectral sensitivity. For fixed material, camera and light spectra, the integral is represented by the albedo,  $\alpha$ :

$$I = \alpha \hat{\mathbf{l}} \cdot \hat{\mathbf{n}}. \quad (2)$$

If the surface is fixed, a minimum of three measurements with non-planar, known lighting directions are required to determine the normal and albedo [16]:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \alpha \begin{bmatrix} \hat{\mathbf{l}}_1 \\ \hat{\mathbf{l}}_2 \\ \hat{\mathbf{l}}_3 \end{bmatrix} \hat{\mathbf{n}} \quad (3)$$

For a dynamic scene, we assume the material reflectance is constant, but the surface normal varies between measurements. The system is now under-constrained:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \alpha \begin{bmatrix} \hat{\mathbf{l}}_1 \cdot \hat{\mathbf{n}}_1 \\ \hat{\mathbf{l}}_2 \cdot \hat{\mathbf{n}}_2 \\ \hat{\mathbf{l}}_3 \cdot \hat{\mathbf{n}}_3 \end{bmatrix} \quad (4)$$

Using a trichromatic camera and three lights of different colors, we measure shading under three different lighting directions simultaneously and thus for a single consistent surface orientation. Consider a camera with three color channels labeled  $r$ ,  $g$ , and  $b$ . Let us assume that each light, indexed by  $j$ , is from direction  $\hat{\mathbf{l}}_j$ , and that each light  $j$  is composed of a weighted combination of a small number of light colors, indexed by  $k$ . For simplicity, first consider a single light of a single color  $k$  and intensity  $w_{kj}$ , and direction  $\hat{\mathbf{l}}_j$ . The pixel intensity of a material illuminated by that light is

$$\mathbf{I} = \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = \begin{bmatrix} \alpha_{kr} \\ \alpha_{kg} \\ \alpha_{kb} \end{bmatrix} w_{kj} \hat{\mathbf{l}}_j^\top \hat{\mathbf{n}} \quad (5)$$

Here,  $(\alpha_{kr}, \alpha_{kg}, \alpha_{kb})^\top$  are the responses of each camera color channel to the material illuminated (from the normal direction) by light of color  $k$ . For example,

$$\alpha_{kr} = \int S_k(\lambda) \rho(\lambda) \nu_r(\lambda) d\lambda. \quad (6)$$

We refer to  $\alpha_{\mathbf{k}} = (\alpha_{kr}, \alpha_{kg}, \alpha_{kb})^\top$  as a vector of “imaging coefficients.” They are not just a property of a specific material; rather, they vary for each different combination of light, material and sensor colors. For a single light of direction  $\hat{\mathbf{l}}_j$  comprised of a linear combination of colors  $k$ , the measured pixel intensity is

$$\mathbf{I} = \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = \left( \sum_k \alpha_{\mathbf{k}} w_{kj} \right) \hat{\mathbf{l}}_j^\top \hat{\mathbf{n}}. \quad (7)$$

For multiple lights, barring occlusions, we sum intensities due to each light:

$$\mathbf{I} = \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = \sum_j \left( \sum_k \alpha_{\mathbf{k}} w_{kj} \hat{\mathbf{l}}_j^\top \right) \hat{\mathbf{n}} = \sum_k \left( \alpha_{\mathbf{k}} \left( \sum_j w_{kj} \hat{\mathbf{l}}_j^\top \right) \right) \hat{\mathbf{n}} = \sum_k \left( \alpha_{\mathbf{k}} \mathbf{l}_{\mathbf{k}}^\top \right) \hat{\mathbf{n}}, \quad (8)$$

where

$$\mathbf{l}_{\mathbf{k}} = \sum_j w_{kj} \hat{\mathbf{l}}_j. \quad (9)$$

Here,  $\mathbf{l}_{\mathbf{k}}$  can be considered the effective direction and intensity of light of color  $k$ . If  $\alpha_{\mathbf{k}}$  are known and linearly independent, and  $\mathbf{l}_{\mathbf{k}}$  are known and linearly independent, then we can measure  $\hat{\mathbf{n}}$  in a single image.

Of course, although the  $\mathbf{l}_{\mathbf{k}}$  may be known in advance for calibrated lights, the reflectance coefficients  $\alpha_{\mathbf{k}}$  for materials in a dynamic scene are generally

<sup>1</sup> Color scientists might cringe at our usage of the words color, red, green, and blue for non-perceptual quantities. For the sake of readability, we use red, green and blue as a shorthand for visible spectra with most of the energy concentrated in longer, medium or shorter wavelengths, respectively. When we say the color of two lights are the same, we mean the spectra are identical up to a scale factor.

unknown. For scenes with spatially varying or unknown materials (and thus unknown  $\alpha_k$ ), we use additional time-multiplexed measurements with changing  $\mathbf{l}_k$ , producing a series of measurements:

$$\begin{aligned}\mathbf{I}^t &= \left( \sum_k \alpha_k \mathbf{l}_k^{t\top} \right) \hat{\mathbf{n}}^t \\ \mathbf{I}^{(t+1)} &= \left( \sum_k \alpha_k \mathbf{l}_k^{(t+1)\top} \right) \hat{\mathbf{n}}^{(t+1)} \\ \mathbf{I}^{(t+2)} &= \left( \sum_k \alpha_k \mathbf{l}_k^{(t+2)\top} \right) \hat{\mathbf{n}}^{(t+2)} \\ &\vdots\end{aligned}\tag{10}$$

We assume, for now, that we can align these measurements perfectly using optical flow. The  $\mathbf{l}_k$  are known in advance, but the  $\alpha_k$  and normals are not. In general, Equation 10 is difficult to solve. If we use  $F$  frames and three light colors ( $k = 3$ ), we have  $9 + 2F$  unknowns for the reflectance coefficients and per-frame normals, but only  $3F$  measurements. We need not, however, recover the surface normal for every frame. Instead, we will use one image with three spatially separated colored lights to measure the surface normal instantaneously, and use additional frames with carefully chosen lighting conditions to recover imaging coefficients  $\alpha_k$  independently of the changing surface orientation.

We consider the minimum of three light colors; using more only adds more unknown imaging coefficients. With three sensor colors and three light colors, Equation 8 can be rewritten as

$$\mathbf{I} = \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = \sum_{k=1}^3 (\alpha_k \mathbf{l}_k) \hat{\mathbf{n}} = \begin{bmatrix} \alpha_{1r} & \alpha_{2r} & \alpha_{3r} \\ \alpha_{1g} & \alpha_{2g} & \alpha_{3g} \\ \alpha_{1b} & \alpha_{2b} & \alpha_{3b} \end{bmatrix} \begin{bmatrix} \mathbf{l}_1^\top \\ \mathbf{l}_2^\top \\ \mathbf{l}_3^\top \end{bmatrix} \hat{\mathbf{n}}.\tag{11}$$

Now we will show that using four images, we can compute the unknown  $\alpha_k$  up to a single global scale factor. We capture three images, each lit by a single color, with the lighting directions being linearly independent and the colors being different for all images. For a point on the moving surface, this yields color pixel intensities  $\mathbf{I}_1$ ,  $\mathbf{I}_2$ , and  $\mathbf{I}_3$ . The image for  $\mathbf{I}_1$  is taken under illumination of color  $k = 1$  with scaled direction  $\mathbf{l}_1 = w_{k1} \hat{\mathbf{l}}_1$ , and so on. We take another image using lights of all three colors from a single direction, producing the follow system:

$$\mathbf{I}_1 = \alpha_1 \mathbf{l}_1^\top \hat{\mathbf{n}}_1 = \alpha_1 s_1 \tag{12}$$

$$\mathbf{I}_2 = \alpha_2 \mathbf{l}_2^\top \hat{\mathbf{n}}_2 = \alpha_2 s_2 \tag{13}$$

$$\mathbf{I}_3 = \alpha_3 \mathbf{l}_3^\top \hat{\mathbf{n}}_3 = \alpha_3 s_3 \tag{14}$$

$$\mathbf{I}_4 = (\alpha_1 + \alpha_2 + \alpha_3) \mathbf{l}_4^\top \hat{\mathbf{n}}_4 = (\alpha_1 + \alpha_2 + \alpha_3) s_4 \tag{15}$$

We have used  $s_1$  to represent the unknown scale factor  $\mathbf{l}_1^\top \hat{\mathbf{n}}_1$  in the first equation, and so on. Solving the top three equations for  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ , respectively, and substituting into the fourth yields

$$\mathbf{I}_4 = (\mathbf{I}_1/s_1 + \mathbf{I}_2/s_2 + \mathbf{I}_3/s_3) s_4, \tag{16}$$

or

$$\mathbf{I}_4 = [\mathbf{I}_1 \ \mathbf{I}_2 \ \mathbf{I}_3] \begin{bmatrix} s_4/s_1 \\ s_4/s_2 \\ s_4/s_3 \end{bmatrix} \quad (17)$$

We solve this system for  $s_1$ ,  $s_2$ , and  $s_3$  up to a scale factor  $1/s_4$ , and use Equations 12, 14 to get  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  up to the same factor. As Equation 8 shows, this ambiguity does not prevent recovery of the normal using a fifth image taken with spatially separated colored lights. In practice, five different images may be too many to capture at video rates, and aligning the set of images may be difficult due to occlusions and varying illumination. In the next section, we explore ways to optimize this method.

## 4 Implementation

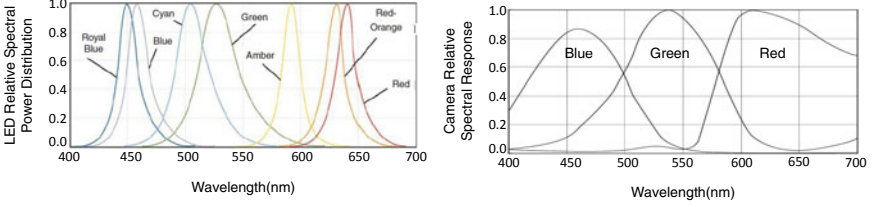
We have presented a theory for instantaneously measuring either surface orientation or imaging properties that vary with the materials. In this section, we investigate using fewer images, facilitating accurate optical flow to align those images, and using commonly available hardware.

*Camera and Light Spectral Characteristics.* A straightforward way to reduce the unknowns at each pixel, and thus require fewer images, is to ensure that some of the imaging coefficients are zero. We might try to use red, green and blue lights such that there is no "crosstalk" between lights and camera color sensors of different colors. Materials illuminated by only the green light, for example, would not register on the camera's red or blue color channels. This corresponds to the simplified component-wise  $(R, G, B)$  imaging model often used in computer graphics and also by a recent work on dynamic photometric stereo using colored lights [3]. Each material and light color is described by an RGB triplet, and the reflected intensity from a Lambertian surface with normal  $\hat{\mathbf{n}}$  and reflectance  $A = (A_R, A_G, A_B)$  lit by light of color  $L = (L_R, L_G, L_B)$  from direction  $\hat{\mathbf{n}}$  is

$$C = (C_R, C_G, C_B) = (A_R L_R, A_G L_G, A_B L_B)(\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}). \quad (18)$$

We explored this approach using a typical single-chip color video camera, the Point Grey Research Flea2 FL2-08S2. With gamma correction off, the Flea2 has a linear response over most of its range. For lighting we use red, green and blue Luxeon K2 light emitting diodes (LEDs). These LEDs are inexpensive, bright, switch quickly (important for TMI), and have relatively narrow spectral power distributions.

Figure 1 shows the spectral characteristics of our camera and LEDs, and reveals two relevant properties. First, the spectra of the blue and green LEDs and the blue and green color sensors significantly overlap. Generally speaking, single-chip color sensors (as well as our own eyes) use color sensors with wide spectral responses for increased sensitivity, so crosstalk is unavoidable (in this



**Fig. 1.** Overlapping LED spectra and camera color responses necessitate using a complete imaging model, not a simplified component-wise RGB one. (Left) Relative spectral power distributions for different color Luxeon K2 LEDs. (Right) Relative spectral response for the red, green and blue pixels on the SONY ICX204 image sensor used in our camera.

case, more than one color sensor responds to the same light color). On the other hand, the red and blue color channels are decoupled; the red LED spectra has virtually no overlap with the blue color sensor, and vice versa. Because we are now assigning color labels like "red" to our lights, we will switch to using capital letters instead of numbers to label light colors. We will use upper case  $R$ ,  $G$ , and  $B$  for light colors, while still using lower case  $r$ ,  $g$ , and  $b$  to for camera color channels. For the decoupled blue and red color channels in our system, we expect  $\alpha_{Rb} = \alpha_{Br} = 0$ . Using images of the patches on a Gretag Macbeth color checker [4] illuminated one color LED at a time, we verified that  $\alpha_{Br}$  and  $\alpha_{Rb}$  are negligible for our hardware. Unfortunately, the crosstalk for the red/green and green/blue color combinations is significant and varies greatly for different materials. We found that  $\alpha_{Bg}/\alpha_{Bb}$  varies across materials from 0.24 to 0.42,  $\alpha_{Gb}/\alpha_{Gg}$  varies from 0.08 to 0.29,  $\alpha_{Rg}/\alpha_{Rr}$  varies from 0.03 to 0.06, and  $\alpha_{Gr}/\alpha_{Gg}$  is on the order of a percent. These ratios are significant and change greatly from patch to patch, meaning that all non-zero imaging coefficients must be measured for any unknown material.

The LED and camera characteristics and the Macbeth experiment suggest an efficient way to eliminate two more imaging coefficients. We place Edmund Optics Techspec 550nm shortpass filters over the green LEDs to block the longer wavelengths sensed by the red camera color sensor, and Thorlabs FB650-40 filters over the red LEDs to ensure that they does not excite the green camera sensor. Now each measured color pixel corresponds to a much simpler equations. For an image taken with illumination from three spatially separated colored lights whose intensities and directions are described by  $\mathbf{l}_R$ ,  $\mathbf{l}_G$ , and  $\mathbf{l}_B$ , Equation 8 yields

$$\begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = \begin{bmatrix} \alpha_{Rr} & 0 & 0 \\ 0 & \alpha_{Gg} & \alpha_{Bg} \\ 0 & \alpha_{Gb} & \alpha_{Bb} \end{bmatrix} \begin{bmatrix} \mathbf{l}_R^\top \\ \mathbf{l}_G^\top \\ \mathbf{l}_B^\top \end{bmatrix} \hat{\mathbf{n}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha'_{Gg} & \alpha'_{Bg} \\ 0 & \alpha'_{Gb} & \alpha'_{Bb} \end{bmatrix} \begin{bmatrix} \mathbf{l}_R^\top \\ \mathbf{l}_G^\top \\ \mathbf{l}_B^\top \end{bmatrix} (\alpha_{Rr} \hat{\mathbf{n}}) \quad (19)$$

Here, we have substituted  $\alpha'_{Gg} = \alpha_{Gg}/\alpha_{Rr}$ ,  $\alpha'_{Bg} = \alpha_{Bg}/\alpha_{Rr}$ , and so on.

We can compute the normal direction from Equation 19 if we know  $\alpha'_{Gg}$ ,  $\alpha'_{Bg}$ ,  $\alpha'_{Gb}$ , and  $\alpha'_{Bb}$ . These values can be measured using just two additional images:

one with red and green lights from the same direction  $\mathbf{l}_{RG}$ , the other with red and blue lights from the same direction  $\mathbf{l}_{RB}$ . The first images gives

$$\begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = \begin{bmatrix} \alpha_{Rr} & 0 & 0 \\ 0 & \alpha_{Gg} & \alpha_{Bg} \\ 0 & \alpha_{Gb} & \alpha_{Bb} \end{bmatrix} \begin{bmatrix} \mathbf{l}_{RG}^\top \\ \mathbf{l}_{RG}^\top \\ \mathbf{0}^\top \end{bmatrix} (\hat{\mathbf{n}}_{RG}) = \begin{bmatrix} \alpha_{Rr} \\ \alpha_{Gg} \\ \alpha_{Bb} \end{bmatrix} (\mathbf{l}_{RG}^\top \hat{\mathbf{n}}_{RG}). \quad (20)$$

Despite the unknown normal  $\hat{\mathbf{n}}_{RG}$ , we can solve for  $\alpha'_{Gg} = \alpha_{Gg}/\alpha_{Rr} = I_r/I_g$  and  $\alpha'_{Gb} = \alpha_{Gb}/\alpha_{Rr} = I_r/I_b$ . Similarly, the second image with red and blue lights from direction  $\mathbf{l}_{RB}$  determines  $\alpha'_{Bg}$  and  $\alpha'_{Bb}$ .

*Time-Multiplexed Illumination and Optical Flow.* Our system uses optical flow to align the two frames for measuring imaging coefficients to the frame illuminated with spatially separated red, green and blue lights. The red light is used for every frame. To facilitate optical flow, we set the red lighting to be constant and from the direction of the camera. Although the green and blue lights vary, they do not affect the red camera sensor, so the red video channel appears to have constant illumination. Setting the red light to arrive from the same direction as the camera prevents any shadows in the red channel of the video. We can robustly estimate optical flow for the red channel between adjacent frames using standard algorithms.

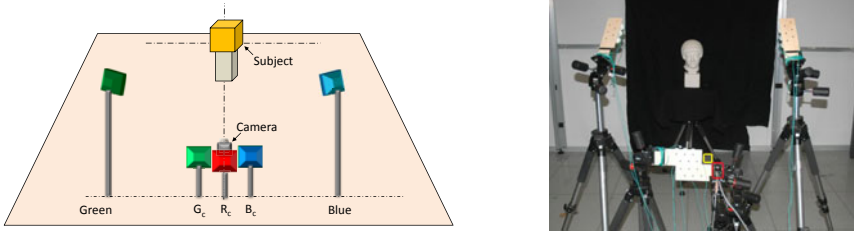
We output orientation measurements at half the video camera frame rate using the following lighting sequence:

$$R_c + G_c \mid R_c + G + B \mid R_c + B_c \mid R_c + G + B \dots$$

Here,  $R_c$ ,  $G_c$ ,  $B_c$ , indicate red, green and blue lights from the direction of the camera, and  $G$  and  $B$  indicate the additional green and blue light directions used to estimate the normal. Each  $R_c + G + B$  image is adjacent to an  $R_c + G_c$  and an  $R_c + B_c$  image.

Because our method measures material properties independently of the surface normal, the optical flow need not be pixel-accurate. As long as the alignment maps regions of the same material to each other, the surface normal estimate will be correct. Segmentation-based optical flow methods, for example, often have this property, even if subtle changes in shading from frame to frame may distort flow estimates within segments of the same material.

*Hardware Design.* Figure 2 shows a schematic of our system and the actual hardware. The setup has three spatially separated red, green and blue lights, labeled  $G$ ,  $B$ , and  $R_c$ . The LEDs are positioned and filtered as described in the previous section. A simple microcontroller circuit triggers the LEDs and camera. We trigger the camera at 30Hz, but compute normal information for a video at half that rate. This is not a fundamental limit of our technique; upgrading to a 60Hz camera would enable normal map computations for a 30Hz sequence. Similar to Hernández et al., we use images of a diffuse plane at multiple known orientations to estimate the light intensities and directions  $\mathbf{l}_G$ ,  $\hat{\mathbf{l}}_{Rc}$ ,  $\mathbf{l}_B$ ,  $\hat{\mathbf{l}}_{Gc}$ , and  $\hat{\mathbf{l}}_{Bc}$ . The lights next to the camera are assumed to have unit intensity, and the



**Fig. 2.** A schematic diagram and the actual hardware used in our system. Each light is made of three LEDs with lenses mounted in a triangle pattern on the wooden boards (some LEDs shown on the boards are not used). The camera (outlined in red) aims through the square notch (shown in yellow) in the top right corner of the bottommost board.

magnitudes of vectors  $\mathbf{l}_G$  and  $\mathbf{l}_B$  specify the intensity ratios between lights  $G$  and  $G_c$ , and  $B$  and  $B_c$ , respectively. For each set of three images, we use the optical flow method of Black and Anandan [2] (but using only the red channel) to compute the location of each point in the  $R_c + G + B$  image in the neighboring  $R_c + G_c$  and  $R_c + B_c$  images. The material imaging coefficients from those points are used to estimate the normals for the  $R_c + G + B$  frame.

## 5 Results

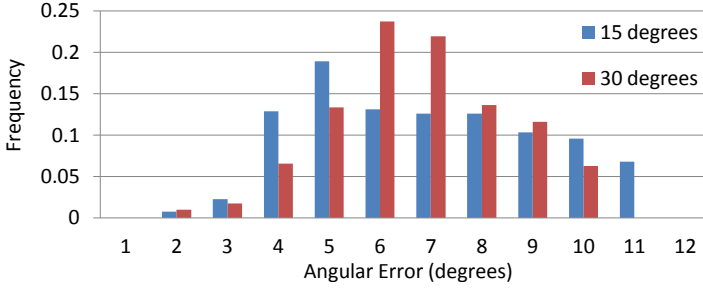
In this section, we present simulations to show the errors caused by (1) assuming constant normals for photometric stereo using alternating white lights, and (2) using a component-wise RGB imaging model in the presence of crosstalk. After that we show surface reconstructions and renderings produced using our method for challenging scenes.

### 5.1 Simulations

*Changing Surface Orientations.* Our first simulation investigates the accuracy of photometric stereo for Lambertian deforming objects using traditional TMI with alternating white lights. We will assume perfect optical flow to align the moving images, so the errors are due purely to the changing surface orientation between measurements. We simulated a system with three alternating white lights, capturing a rotating white surface with albedo  $\alpha = 1.0$ . The three measurements are the dot product of the normal and lighting directions:  $I_1 = \hat{\mathbf{l}}_1 \cdot \hat{\mathbf{n}}_1$ ,  $I_2 = \hat{\mathbf{l}}_2 \cdot \hat{\mathbf{n}}_2$ , and  $I_3 = \hat{\mathbf{l}}_3 \cdot \hat{\mathbf{n}}_3$ . Combining these observations and assuming a constant normal is equivalent to solving the system  $\mathbf{I} = \mathbf{L}\hat{\mathbf{n}}_c$ , where the rows of  $\mathbf{L}$  are  $\hat{\mathbf{l}}_1$ ,  $\hat{\mathbf{l}}_2$ , and  $\hat{\mathbf{l}}_3$ ; and  $\mathbf{I} = (I_1, I_2, I_3)^\top$ .

We simulated a 30fps camera pointing along the negative  $z$  axis, viewing a surface at the origin with 1Hz rotational motion. 1Hz is actually a conservative number; people often turn their heads, hands or fingers at this rate. Many interesting performances such as dancing or martial arts involve rotational deformations that are much more rapid. We used three lighting directions,  $15^\circ$  from



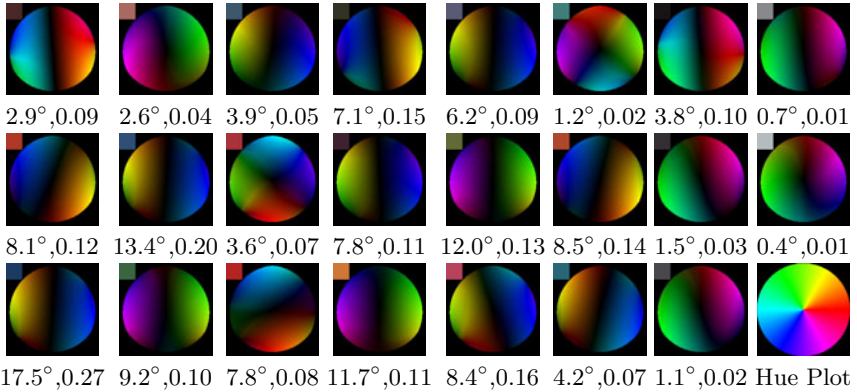


**Fig. 3.** Traditional photometric stereo using alternating white lights errs if the normal is changing. Here we show a histogram of the angular errors for estimated surface orientations using three alternating white lights, a 30fps camera, and a surface with 1Hz rotational deformation. We evenly sampled a range of surface orientations and rotational axes, for light directions  $15^\circ$  and  $30^\circ$  off the z axis. In both cases, we see a broad distribution of angular errors as high as  $10^\circ$ .

and evenly spaced around the z axis, and then repeated the simulations with the lighting  $30^\circ$  off the z axis. The surface normal for the middle frame was a vector  $(0, 0, 1)$  pointing at the camera and rotated up or down (i.e. about the y axis) anywhere from  $-50^\circ$  to  $50^\circ$ , in  $10^\circ$  increments. To simulate object motion, this normal rotated backward and forward  $12^\circ$  (for 30Hz rotation filmed with a 30fps camera) to generate the first and third measurements. We also changed the axis of rotation itself, using axes in the x-y plane, evenly spaced from  $0^\circ$  to  $360^\circ$  in  $10^\circ$  increments.

Figure 3 shows a histogram of the angular error between the true and computed surface normals for the middle frame. For the  $15^\circ$  off-axis lights, the mean and standard deviation of the angular error is  $5.9^\circ$  and  $2.3^\circ$ . For the  $30^\circ$  off-axis lights, the mean and standard deviation of the angular error is  $5.7^\circ$  and  $1.7^\circ$ . The computed normals are not simply averages of the observed ones; because of the varying lighting directions, even for a surface normal rotating in a plane, the computed orientation may not lie in the same plane. Orientation errors are also accompanied by reflectance errors. The mean albedo error (relative to the ground truth of 1.0) was 0.032 for light directions at  $15^\circ$  to the z axis, and 0.042 for  $32^\circ$ , with standard deviations of 0.039 and 0.050. Of course, these errors might change for different parameters. Regardless, they are a fundamental accuracy limit for dynamic photometric stereo with TMI if one ignores the time-varying normal.

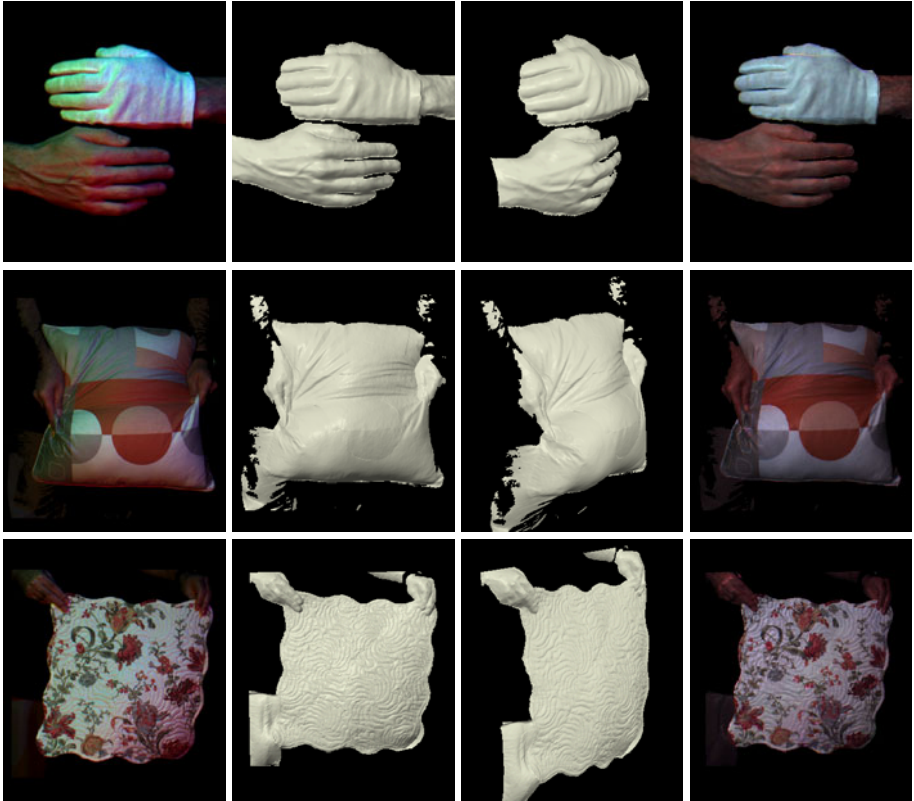
*RGB Component-Wise Imaging Models.* Our second simulation investigated the errors from using a component-wise RGB camera for photometric stereo in the presence of crosstalk. In practice, such a system would alternate between red, green and blue light from a single direction in order to measure material reflectances, and spatially separated lights to measure surface orientation. We implemented the RGB component-wise imaging model using actual imaging data



**Fig. 4.** Photometric stereo using colored lights with a simplified component-wise ( $R, G, B$ ) model causes inaccurate normal and reflectance estimates. This simulation used red, green and blue lights  $15^\circ$  off the  $z$ -axis. These visualizations show hue plots of the surface normal directional error for spheres with colors corresponding to the Macbeth color checker patches. We show each patch’s imaged color (inset squares), the maximum angular error (degrees) for estimated normals over the sphere, and the maximum albedo error (defined as the error for the computed normal’s length, which should be 1.0). The white patch, used to fit the model, had negligible error.

for the Macbeth color checker and our LEDs and Flea2 camera, and simulated a stationary object (so these are ideal results). We took three pictures of the Macbeth chart illuminated by a single red, green, or blue LED, and computed the coefficient matrix  $\mathbf{M}$  for each light, material and sensor combination. We let the color of the white Macbeth checker be  $(1, 1, 1)$  and used the component-wise model to compute the color of each light, and then of all the checkers. We used the real-world imaging data to simulate Lambertian reflection off a sphere illuminated by a red, a green, and a blue light from  $15^\circ$  off the  $z$  axis, as before. To be conservative, we only simulated surface normals at angles less than  $85^\circ$  from all three lights.

Figure 4 shows that the angular orientation error using the component-wise model can be quite large. For the white patch, the simplified model works perfectly. The other patches show a fairly even spread of errors from nearly zero for the other grayscale patches to as high as  $17^\circ$ . The computed normals are all accurate at  $(0, 0, 1)$  because the lights in these simulations are evenly spaced around the  $z$  axis, so the Lambertian shading terms are all equal at that one point. The clear axes in the error visualizations are due to our system having strong crosstalk between only two of the three color channels. These data show that using the component-wise imaging model leads to large surface orientation errors, especially at oblique angles. By contrast, our system, without sacrificing frame rate (i.e. still computing normals for every other input frame), computes accurate orientation despite significant crosstalk in two color channels.



**Fig. 5.** Results from our photometric stereo system. From top to bottom, the rows are (1) a video frame illuminated with spatially separated red, green and blue LEDs (2) reconstructed geometry (3) geometry rendered from a new view, and (4) geometry rendered from the camera view and textured with measured appearance data. The hands are rotating around the axis of the arms, the pillow is being creased, and the mat is being waved. The pillow shows that we are computing consistent normals despite changing material colors. The artifacts at color edges are due to resampling during image alignment. We recover the fine quilted surface detail of the mat well despite its colorful pattern. The supplemental material video shows the entire input and output video sequences.

## 5.2 Dynamic Photometric Stereo with Real Objects

Figure 5 shows three results using our system capture the shape and appearance of different moving and deforming objects. After computing surface normals, we reconstructed the surface geometry by integrating the surface normals, equivalent to solving a Poisson equation [12]. As the images show, we recovered the fine detail on the hands and glove (veins, wrinkles, etc.) well. The geometry for the creasing pillow is consistent despite sudden color changes. The mat is particularly interesting; the color texture is very complicated and prominent,

yet barely detectable in the recovered geometry. The reader is encouraged to view the supplemental videos showing the motion in the input images, the reconstructed geometry, and the textured geometry for all three sequences. The hands rotate at roughly 1/6Hz, and the fingers curl even more rapidly.

## 6 Discussion

The fundamental goal of photometric stereo for moving, non-rigid objects is to estimate time-varying surface orientations. The prior art using TMI, however, assumes constant surface orientations across the frames used for the estimates, fundamentally limiting their accuracy. By contrast, our time and color multiplexed photometric stereo method is the first that is robust to changing surface orientations for non-rigid scenes. We have shown that for Lambertian surfaces and general imaging models, five images with appropriately chosen lighting are sufficient to recover the time-varying surface orientation for one frame. Our optimized implementation requires only three images. Because our method measures reflectance coefficients independently of the changing surface orientations, it preserves the key strength of colored lights for photometric stereo: an instantaneous orientation estimate in one frame, given known material reflectances.

Like the prior art, we use optical flow to align measurements from several video frames. Shading changes due to the deforming surfaces may complicate this alignment. Even in the ideal case of perfect optical flow, however, time-varying surface normals lead to errors for the prior art. By contrast, our method does not even require pixel accurate alignment. As long as surfaces of the same material are aligned to each other, the imaging coefficients are estimated correctly. Our implementation not only tolerates less than pixel-accurate alignment, but also makes the alignment more robust by fixing the apparent illumination for video in the camera red color channel. Using standard optical flow methods for that channel alone, we can directly align the frames used to measure material properties to the one with spatially separated lights for the orientation estimates. We need not assume linear motion across multiple frames, nor capture extra images to serve as optical flow key frames.

Our system is simple, consisting of an ordinary camera and LEDs with filters, yet captures detailed shapes of moving objects with complicated color textures. Like other three-source photometric stereo methods, it can err in the presence of non-Lambertian reflectance, interreflection, occlusions and mixed pixels. One might argue for a system with three completely isolated color channels (a red sensor that only responds to a red light, and so on). In addition to being difficult to implement in practice, such a design has another drawback: it cannot measure materials with no reflectance in one or more of the color channels. Our implementation suffers such limitations, but to a lesser extent. Saturated colors are not uncommon. Because our theory assumes a general imaging model with crosstalk, the five image solution could be used with broad-spectrum light colors and color sensors to capture the shapes of materials with a very wide range of colors (the constraint is that the three  $\alpha_{\mathbf{k}}$  must be linearly independent).

## References

1. Basri, R., Frolova, D.: A two-frame theory of motion, lighting and shape. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
2. Black, M., Anandan, P.: A framework for the robust estimation of optical flow. In: IEEE International Conference on Computer Vision, pp. 231–236 (1993)
3. Decker, B.D., Kautz, J., Mertens, T., Bekaert, P.: Capturing multiple illumination conditions using time and color multiplexing. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
4. Gretag Macbeth Color Management Solutions, <http://www.gretagmacbeth.com>
5. Hernández, C., Vogiatzis, G., Brostow, G.J., Stenger, B., Cipolla, R.: Non-rigid photometric stereo with colored lights. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
6. Joshi, N., Kriegman, D.: Shape from varying illumination and viewpoint. In: IEEE International Conference on Computer Vision (2007)
7. Kontsevich, L.L., Petrov, A.P., Vergelskaya, I.S.: Reconstruction of shape from shading in color images. *J. Opt. Soc. Am. A* 11(3), 1047–1052 (1994)
8. Moses, Y., Shimshoni, I.: 3d shape recovery of smooth surfaces: Dropping the fixed viewpoint assumption. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 429–438. Springer, Heidelberg (2006)
9. Nehab, D., Rusinkiewicz, S., Davis, J., Ramamoorthi, R.: Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. on Graphics* 24(3), 536–543 (2005)
10. Petrov, A.: Light, color and shape. *Cognitive Processes and their Simulation*, 350–358 (1987) (in Russian)
11. Simakov, D., Frolova, D., Basri, R.: Dense shape reconstruction of a moving object under arbitrary, unknown lighting. In: IEEE International Conference on Computer Vision (2003)
12. Simchony, T., Chellappa, R., Shao, M.: Direct analytical methods for solving poisson equations in computer vision problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12(5), 435–446 (1990)
13. Vlasic, D., Peers, P., Baran, I., Debevec, P., Popović, J., Rusinkiewicz, S., Matusik, W.: Dynamic shape capture using multi-view photometric stereo. *ACM Trans. on Graphics* 28(5) (2009)
14. Weise, T., Leibe, B., Gool, L.V.: Fast 3d scanning with automatic motion compensation. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
15. Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., Debevec, P.: Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. on Graphics* 24(3), 756–764 (2005)
16. Woodham, R.: Photometric method for determining surface orientation from multiple images. *Optical Engineering* 19(1), 139–144 (1980)

# Fully Isotropic Fast Marching Methods on Cartesian Grids

Vikram Appia and Anthony Yezzi

Georgia Institute of Technology, GA, USA

**Abstract.** The existing Fast Marching methods which are used to solve the Eikonal equation use a locally continuous model to estimate the accumulated cost, but a discontinuous (discretized) model for the traveling cost around each grid point. Because the accumulated cost and the traveling (local) cost are treated differently, the estimate of the accumulated cost at any point will vary based on the direction of the arriving front. Instead we propose to estimate the traveling cost at each grid point based on a locally continuous model, where we will interpolate the traveling cost along the direction of the propagating front. We further choose an interpolation scheme that is not biased by the direction of the front. Thus making the fast marching process truly isotropic. We show the significance of removing the directional bias in the computation of the cost in certain applications of fast marching method. We also compare the accuracy and computation times of our proposed methods with the existing state of the art fast marching techniques to demonstrate the superiority of our method.

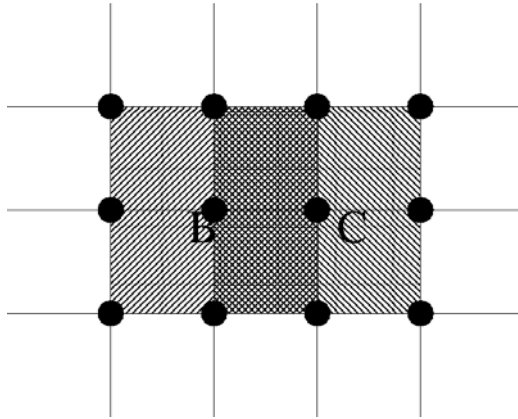
**Keywords:** Fast Marching Methods, Isotropic Fast Marching, Segmentation, Tracking, FMM, Eikonal Equation, minimal cost path.

## 1 Introduction

A large number of computer vision applications such as segmentation, tracking, optimal path planning *etc.* use the minimal cost path approach. The Fast Marching Method which is widely used to solve the minimal path problem was first introduced by Sethian [1,10] and Tsitsiklis [11]. Cohen and Kimmel [4,5] later noticed that the minimal cost problem satisfies the Eikonal equation,

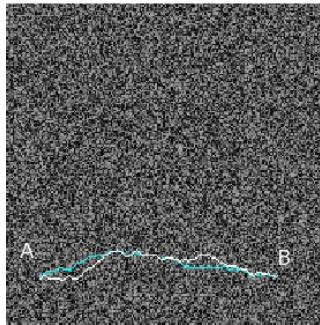
$$\|\nabla u\| = \tau. \quad (1)$$

For the Eikonal equation [1] defined on a Cartesian Grid,  $\tau(x)$  would be the traveling cost at a given grid point and  $u(x)$ , the accumulated cost. Since we solve the Eikonal equation numerically on Cartesian Grids, it is impossible to find the exact solution. Some modifications have been suggested in [6,7] to improve the accuracy of the Fast Marching method. Authors in [6,8,9,11] also suggest using an 8-connected neighbor scheme to improve accuracy. All these techniques use a locally continuous model to estimate the accumulated cost, but assume the traveling cost to be constant (discretized) around each grid point. Only [6] interpolates  $\tau$  by shifting it to the center of the grid with a nearest neighbor interpolation, but it still assumes a discretized shifted grid for  $\tau$ . In this paper we propose to use a locally continuous model to estimate  $\tau$  as well.



**Fig. 1.** Overlap in the influence areas of  $\tau_B$  and  $\tau_C$

For the geometry shown in Figure 1 the Fast Marching Method uses linear approximation to compute the accumulated cost at the point  $C$ , but it uses a constant traveling cost  $\tau_C$  for each of the four grid cells containing the point  $C$ . The influence area of the cost function given at a grid point will include all the four quadrants around it. Thus, there is an overlap in the areas of influence of the grid points  $B$  and  $C$ . This means the value of  $u_C$  will vary depending on the direction from which the front is arriving. Ideally, for isotropic fast marching, the accumulated cost should be independent of the direction of the arriving front. For the image shown in Figure 2 we use the traveling cost,  $\tau(x) = I(x)$ , where  $I(x)$  is the intensity at each pixel. The accumulated cost in traveling from point  $A$  to  $B$  should be equal to the cost in traveling from  $B$  to  $A$ . But, due to the dependence on the direction of marching, there will be a difference in the accumulated costs. Figure 2 compares the minimal path obtained using back propagation from end point  $B$  to the source point  $A$  with the minimal path obtained by reversing the direction of front propagation. The difference in the two paths highlights the error caused by the directional dependence of the Fast Marching method.

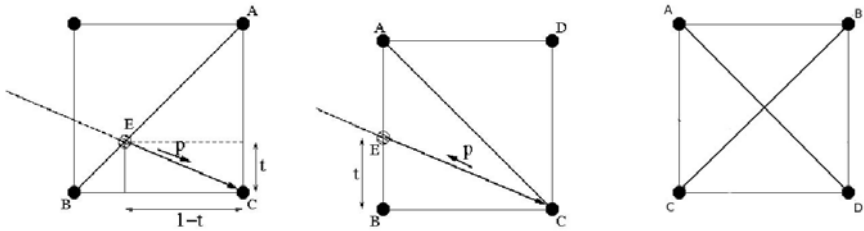


**Fig. 2.** Image with random noise

In this paper we propose two methods to overcome the above-mentioned shortcomings. The first method uses a linear/bilinear model locally to estimate  $\tau$  along the direction of the propagating front within each grid cell. Here we use a continuous model to estimate  $\tau$  and also take the direction of arrival into consideration. We also discuss how the scheme can be made truly isotropic by removing any bias due to the marching direction. We call this method the Interpolated Fast Marching Method and it is discussed in detail in Section 2. In the second method we calculate  $u$  on an upsampled grid. In upsampling the grid,  $\tau$  in the neighborhood of each grid point becomes constant, which eliminates the need to estimate  $\tau$  using a continuous model. We will use the value of  $\tau$  from the direction of arriving front. The upsampled version of the 4 and 8-connected neighbor schemes are discussed in Section 3. Finally, in Section 4 we describe a few numerical experiments conducted to highlight the significance of making the fast marching method independent of direction and we test the accuracy of the proposed methods.

## 2 Interpolated Fast Marching Method

For interpolated Fast Marching scheme we will assume  $\tau$  to be continuous around each grid point and use linear/bilinear interpolation to estimate the value of the local traveling cost within each grid cell. Here we will derive the equations for the linear and bilinear Interpolated Fast Marching schemes. To estimate the traveling cost in a grid cell, the bilinear scheme will use the value of  $\tau$  from all the grid points for a given quadrant. Since only 2 neighbors are used in each quadrant to calculate  $u$  in a 4-connected neighbor scheme, we only discuss the 8-connected neighbor scheme with bilinear interpolation.



(a) 4-Connected Neighbors Scheme (b) 8-Connected Neighbors Scheme (c) Isotropic triangulation of a Grid Cell

Fig. 3. Triangulation of Grid cells

### 2.1 Linear Interpolation

**4-Connected Neighbors Scheme.** Consider a front arriving at the grid point  $C$  from the quadrant  $AB$  and intersecting  $\overline{AB}$  at  $E$  as shown in Figure 3(a). We will use the linear interpolation of the local traveling cost along the path  $\overrightarrow{EC}$  to compute  $u_C$ . Thus the accumulated cost at  $C$  will be,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_B(1-t) + u_A t + \int_0^1 \tau(p) \sqrt{t^2 + (1-t)^2} dp \right\}. \quad (2)$$



Substituting,  $\tau(p) = \tau_C + (\tau_A - \tau_C)p(1-t) + (\tau_B - \tau_C)pt$ ,  $0 \leq p \leq 1$ , in (2) we get,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_B(1-t) + u_A t + \sqrt{t^2 + (1-t)^2} \left( \frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right) \right\}. \quad (3)$$

We get the necessary optimality condition to obtain the minimum of  $u_C$  by solving  $\frac{du_C}{dt} = 0$ , which yields,

$$\begin{aligned} & u_A - u_B + \sqrt{t^2 + (1-t)^2} \left( \frac{\tau_B - \tau_A}{2} t \right) \\ & + \frac{2t-1}{\sqrt{t^2 + (1-t)^2}} \left( \frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right) = 0. \end{aligned} \quad (4)$$

**8-Connected Neighbors Scheme.** The geometry for 8-connected neighbors is shown in Figure 3(b). Using linear interpolation to estimate the local traveling cost along  $\vec{EC}$ , the accumulated cost,  $u_C$ , will be,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_B(1-t) + u_A t + \int_0^1 \tau(p) \sqrt{1+t^2} dp \right\}. \quad (5)$$

Substituting,  $\tau(p) = \tau_C + (\tau_B - \tau_C)p + (\tau_A - \tau_B)pt$ ,  $0 \leq p \leq 1$ , in (5) we get,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_A t + u_B(1-t) + \sqrt{1+t^2} \left( \frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{2} t \right) \right\}. \quad (6)$$

Again the minimizer of  $u_C$  can be obtained by solving  $\frac{du_C}{dt} = 0$ . Thus we have,

$$u_A - u_B + \sqrt{1+t^2} \left( \frac{\tau_A - \tau_B}{2} \right) + \frac{t}{\sqrt{1+t^2}} \left( \frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{2} t \right) = 0. \quad (7)$$

**Isotropic linear interpolation scheme.** Figure 3(a) and 3(b) show the triangulation of a grid cell for the 4 and 8 neighbor schemes respectively. Depending on the front direction one of the quadrant/octant will be chosen to estimate the accumulated cost. But this will induce a directional bias. To overcome this directional bias, we will have to consider all possible triangulations shown in Figure 3(c). In effect the accumulated cost across a grid cell must be the minimum of the solutions obtained using the 4 and 8 neighbor schemes. This would make the scheme completely unbiased to direction and we call this scheme the Iso-Linear scheme.

## 2.2 Bilinear Interpolation

**8-Connected Neighbors Scheme.** The bilinear interpolation to estimate the local traveling cost along  $\vec{EC}$  is given by,

$$\tau(p) = \tau_A(p)(pt) + \tau_B(p)(1-pt) + \tau_C(1-p)(1-pt) + \tau_D(1-p)(pt).$$

It is inherently independent of any directional bias within a grid cell. Substituting, this value of  $\tau(p)$  for  $0 \leq p \leq 1$ , in (5) we get,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_A t + u_B (1-t) + \sqrt{1+t^2} \left( \frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{3} t + \frac{\tau_D - \tau_C}{6} t \right) \right\}. \quad (8)$$

We will again solve  $\frac{du_C}{dt} = 0$ , which yields,

$$\begin{aligned} u_A - u_B + \sqrt{1+t^2} \left( \frac{\tau_A - \tau_B}{3} + \frac{\tau_D - \tau_C}{6} \right) \\ + \frac{t}{\sqrt{1+t^2}} \left( \frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{3} t + \frac{\tau_D - \tau_C}{6} t \right) = 0. \end{aligned} \quad (9)$$

Algebraic manipulations on (4), (7) and (9) will yield quartic equations. We used the Ferrari and Newton methods to solve these quartic equations. We compared the solutions from both techniques and found that they generate equally accurate solutions. Since Newton's method has a quadratic convergence, three iterations were sufficient for convergence. Fixing the number of iterations in each update step also ensures that we have the same computation complexity in each update. This makes the technique suitable to implement on hardware. The solution to Newton's method has fewer (logical and mathematical) operations in comparison to finding the Ferrari (analytic) solution; hence using Newton's method is computationally efficient. We compare the computation times of the two methods on a 500x500 grid in the Table 1. Here we call the 4 and 8-connected neighbor linear Interpolated Fast Marching schemes, Linear-4 and Linear-8 respectively and the 8-connected neighbor bilinear Interpolated Fast Marching scheme, Bilinear-8. The computation times were measured on a laptop with a 1.73 GHz Processor.

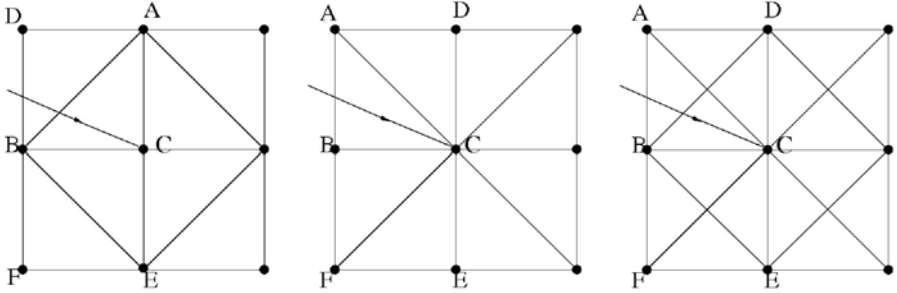
**Table 1.** Comparison of computation times

	Linear-4	Linear-8	Bilinear-8
Analytic (Ferrari)	1.51s	2.83s	3.23s
Newton's Method	0.51s	0.52s	0.65s

### 2.3 Marching Forward Loop

We will still follow the main loop as explained in the basic Fast Marching method [10]. But, when a *trial* point is *accepted* in the min heap structure we will compute the value of  $u$  from both the quadrants/octants which include the newly *accepted* point and replace the newly calculated  $u$  with the minimum of the two solutions and the existing value of  $u$  (if the point is marked as *trial*).

Consider the example in Figure 4(a) where  $B$  is the newly *accepted* point and the accumulated cost at neighbor  $C$  is to be computed. As opposed to the basic fast marching technique,  $u_C$  does not solely depend on  $u_A, u_B, u_E$  and the local traveling cost,  $\tau_C$ , but it also depends on the costs at all the other 8-connected neighbors. Thus, using



(a) 4-Connected Neighbors Scheme (b) 8-Connected Neighbors Scheme (c) Isotropic Fast Marching Scheme

**Fig. 4.**  $B$  is the newly *accepted* grid point and  $u_C$  is to be computed

the quadrant containing the minimum of  $u_A$  and  $u_E$  will not necessarily guarantee the minimum solution to (3). Hence we have to consider both the quadrants that contain  $B$ . If the front also arrives at  $C$  from the other two quadrants, they will be considered when the corresponding neighbors become *accepted*. The same argument can be extended to the 8-connected neighbor case shown in Figure 4(b). Here we only need to calculate  $u_C$  from the two octants containing  $\overline{AB}$  and  $\overline{FB}$  once point  $B$  is *accepted*. For the front arriving at point  $C$  as shown in Figure 2(c), we will consider the possibilities of the front arriving from  $\overline{AB}, \overline{BD}$  and  $\overline{DA}$ .

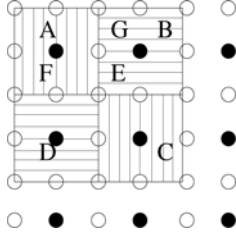
We depart from the traditional Fast Marching method only in the update procedure for the accumulated cost, but follow the same main (outer) loop. Thus the parallel algorithm explained in Bronstein et al. [2], can be extended for the implementation on hardware.

### 3 Upsampled Fast Marching Method

Figure 5 shows that there is no overlap in the influence areas of  $\tau$  on the upsampled grid. Here the solid circles are the grid points from the original grid. Since the traveling cost is constant in each grid cell, there is no directional bias in the calculation of  $u$ . We will compute  $u$  on the upsampled grid and then downsample the output on the original grid.

#### 3.1 4-Connected Neighbors Scheme

In the upsampled grid,  $\tau$  is constant in each quadrant around a grid point. Again the constant traveling cost within each grid cell makes this scheme isotropic. Depending on the direction of the front we will choose the value of  $\tau$  in calculating  $u$ . For example, if the front arrives at  $E$  from the north-west then we would use  $\tau_A$  (Figure 5). At the point  $G$  we would use  $\tau_A$  for a front arriving from the west and  $\tau_B$  for a front arriving from the east. We would use  $\tau_A$  to calculate  $u_A$  irrespective of the direction of the arriving front. Since the value of  $\tau$  is constant along the direction of the front at a sub-pixel



**Fig. 5.** No overlap in the influence areas of  $\tau_A$ ,  $\tau_B$ ,  $\tau_C$  and  $\tau_D$

level, it is not necessary to assume a locally continuous model in interpolating  $\tau$ . Thus, the accumulated cost at  $E$  with the front arriving from the north-west would be,

$$u_E = \min_{0 \leq t \leq 0.5} \left\{ u_F t + u_G (0.5 - t) + \tau_A \sqrt{t^2 + (0.5 - t)^2} \right\} \quad (10)$$

This minimization leads to the closed form solution,

$$u_E = \begin{cases} \frac{(u_F + u_G + \sqrt{\delta})}{2} & \text{if } \delta \geq 0 \\ \min(u_F, u_G) + \frac{\tau_A}{2} & \text{otherwise} \end{cases}$$

where,  $\delta = \frac{\tau_A^2}{2} - (u_F - u_G)^2$ .

### 3.2 8-Connected Neighbors Scheme

As in the case with 4-connected neighbors,  $\tau$  is constant in each octant around a grid point in the upsampled grid. We note that there will be exactly one point in each octant that corresponds to a point in the original grid. We will use the corresponding value of  $\tau$  to compute  $u$ .

By following the procedure described in Section 2.3, we calculate  $u$  only from the two octants that contain the newly *accepted* point. If  $F$  is the newly *accepted* point, we will calculate  $u_E$  in the octants containing  $\overline{FA}$  and  $\overline{FD}$  (Figure 5). The solution will be the minimum of the two values obtained. Thus, for a front arriving from north-west, the accumulated cost at  $E$  will be,

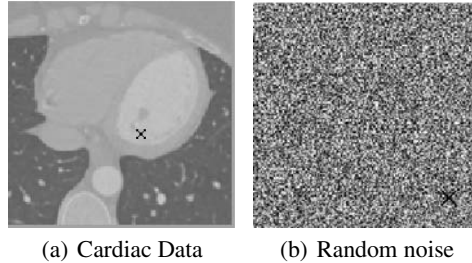
$$u_E = \min_{0 \leq t \leq 0.5} \left\{ u_A t + u_F (0.5 - t) + \tau_A \sqrt{0.5 + t^2} \right\} \quad (11)$$

giving the closed form solution,

$$u_E = \begin{cases} u_F + \frac{\tau_A}{2} & \text{if } u_F \leq u_A \\ u_A + \sqrt{2} \frac{\tau_A}{2} & \text{if } \tau_A \leq 2\sqrt{2}(u_F - u_A) \\ u_F + \frac{\sqrt{\tau_A^2 - 4(u_F - u_A)^2}}{2} & \text{otherwise} \end{cases}$$

## 4 Numerical Experiments

We conducted a few experiments to compare the proposed methods to the basic Fast Marching Method (FMM) [10], Tsitsiklis scheme [11], Shifted-Grid Fast Marching (SGFM) [6] and Multi-stencil Fast Marching (MSFM) [7]. We also compare the upsampled 4 and 8-connected neighbor Fast Marching schemes with the upsampled version of the SGFM scheme (upSG).

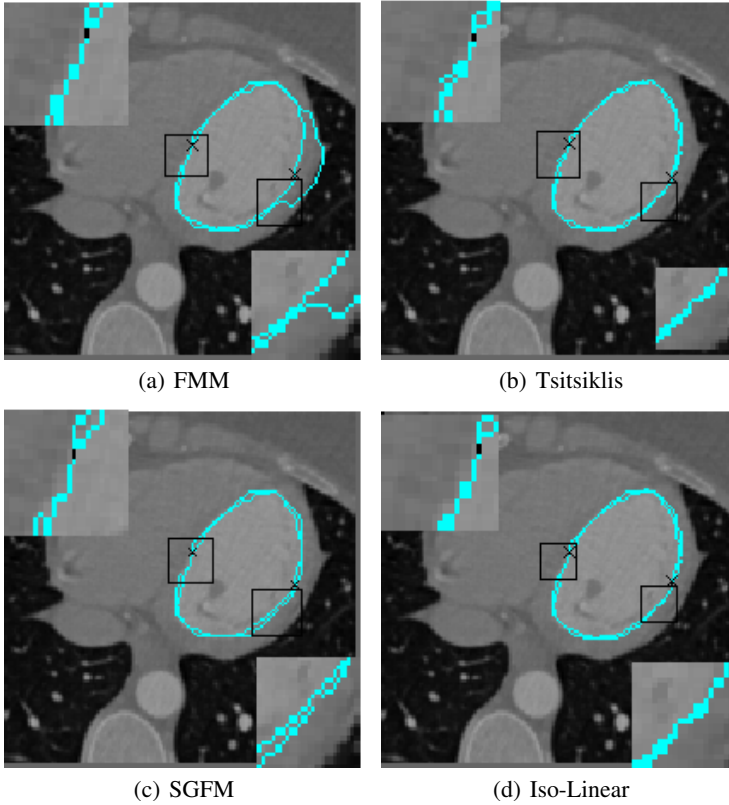


**Fig. 6.** Test Images

In the first experiment we pick a random point, marked by the ‘x’ in the images shown in Figure 6, and compute  $u$  at every point of the image. We then compute the total cost in propagating a front from each point of the image back to the point marked by the ‘x’. We take the average of the difference (error) across the entire image. The numerical values are listed in the Table 2, under the column labeled Average Back Propagation Error (ABPE). We used the cost function,  $\tau(x) = \frac{1}{1+|\nabla J|^2}$  for the cardiac image and  $\tau(x) = I(x)$  for the random noise image.

In Figure 7 we present the results of segmenting the left ventricles in a 2D cardiac slice. To segment the image we pick a point on the boundary of the object and compute the saddle points as described in [5]. From each saddle point we then obtain two minimal paths back to the initial point; these paths will give the segmentation of the object. The minimal paths were obtained using a sub-pixel level back propagation scheme. We then choose the saddle point which minimizes the Chan-Vese [3] energy of the obtained segmentation. Images in Figure 7 show the overlay of segmentation curves initialized with 2 different user given points on the boundary. We see that the segmentation curves are not consistent and they depend on the initialization. This is mainly due to the difference in the marching direction in each case and weak image features at certain locations. We highlight certain regions in these images to compare the segmentation obtained from the different methods.

In the images shown in Figure 8, we compare the minimal paths obtained in traveling from point ‘0’ to points ‘1’, ‘2’ and ‘3’ with the corresponding paths obtained by reversing the direction. We see that using interpolated FMM gives consistent paths, even in the absence of any strong image feature. The results are in accordance to the Average Back Propagation Errors listed in Table 2. The ABPE for the Tsitsiklis scheme is the highest and accordingly the paths obtained with the Tsitsiklis scheme show a lot of variation. Although the SGFM shows lower average error there are variations in



**Fig. 7.** A comparison of segmentation

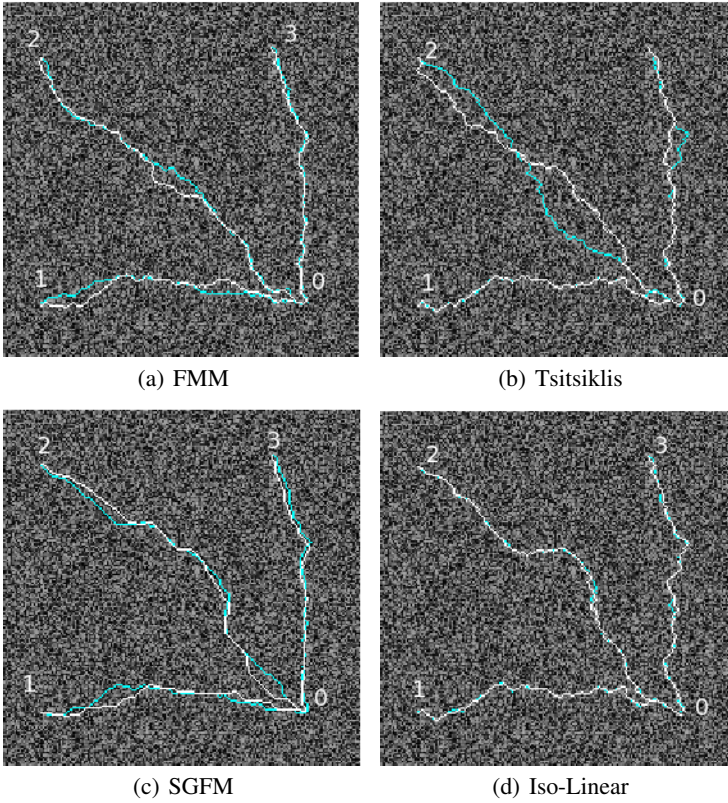
the obtained minimal paths. This is because the interpolation of the cost function in SGFM is equivalent to image smoothing for the  $\tau$  ( $\tau(x) = I(x)$ ) used in this example. This decreases the corresponding average error, but it also decreases the difference in the geodesic distances of the various paths. Thus with the change in the marching direction, the back propagation takes different paths between two given points.

In the next example we compare the accuracy of the various techniques for two cost functions on a  $50 \times 50$  grid,

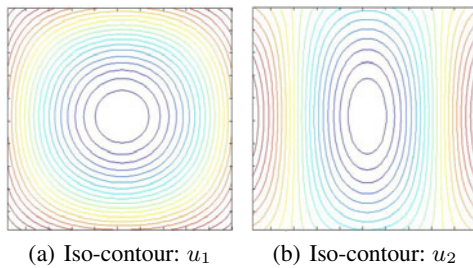
$$\begin{aligned}\tau_1(x, y) &= 1/20 \sqrt{(\sin \frac{x}{20} \cos \frac{y}{20})^2 + (\cos \frac{x}{20} \sin \frac{y}{20})^2}, \\ \tau_2(x, y) &= 1/10 \sqrt{(\sin \frac{x}{20} \cos \frac{y}{10})^2 + (\cos \frac{x}{20} \sin \frac{y}{10})^2}.\end{aligned}$$

The iso-contours of  $u_{analytic}$  are shown in Figure 9. The geodesics from the center  $(26, 26)$  of the grid will be straight lines for  $\tau_1$  and curved for  $\tau_2$ . Since, we have the analytic solution for these cost functions, we can compare the  $L_1$ ,  $L_2$  and  $L_\infty$  norms for each method.

$$\begin{aligned}L_1 &= \text{mean}(|u - u_{analytic}|), \\ L_2 &= \text{mean}(|u - u_{analytic}|^2), \\ L_\infty &= \max(|u - u_{analytic}|).\end{aligned}$$



**Fig. 8.** A comparison of tracking



**Fig. 9.** Iso-contours

The numerical errors in using cost functions  $\tau_1$  and  $\tau_2$  are listed in Table 2. Notice that the error norms show significant improvement for the proposed methods, especially in the case with curved geodesics ( $\tau_2$ ). The iso-contours of the errors for  $\tau_2$  while using FMM, SGFM, Iso-Linear and up8 are shown in Figure [10](#).

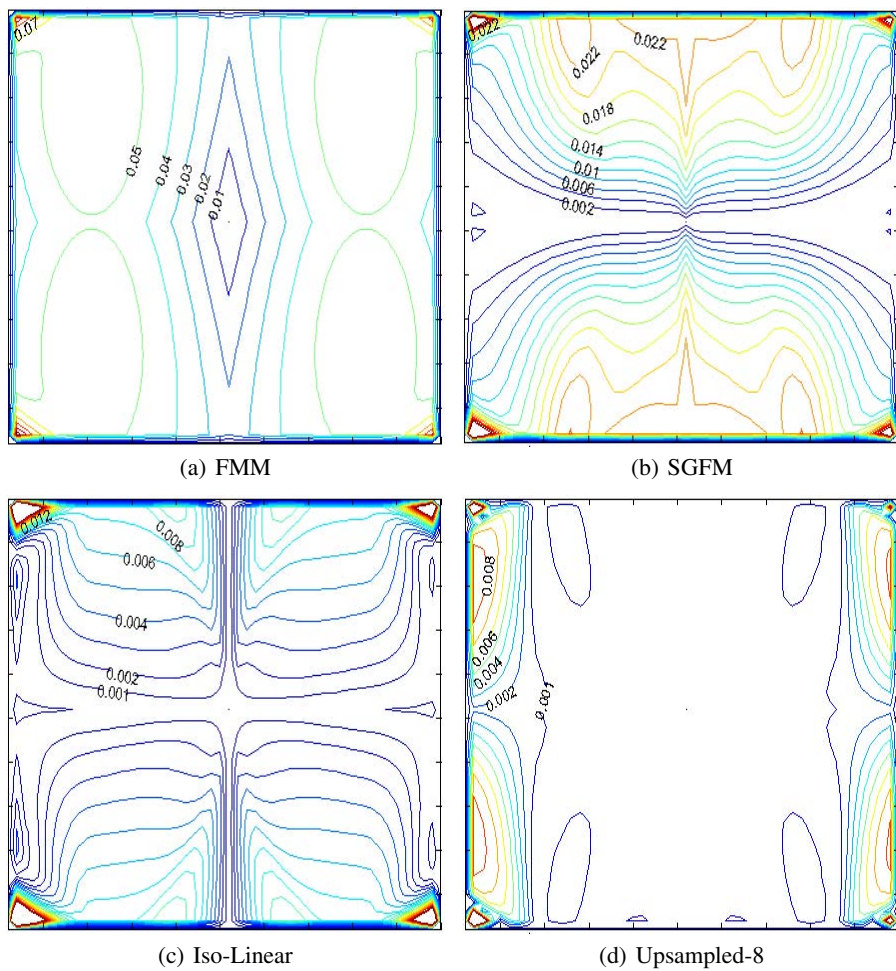


Fig. 10. Iso-contours of errors for  $\tau_2$



**Table 2.** Error norms for  $\tau_1$  and  $\tau_2$ , Average Back Propagation Errors and Computation times

	$\tau_1$			$\tau_2$			ABPE		Time (s)
	$\bar{L}_1$	$\bar{L}_2$	$\bar{L}_\infty$	$\bar{L}_1$	$\bar{L}_2$	$\bar{L}_\infty$	$\bar{I}_1$	$\bar{I}_2$	
FMM	$2.46 \times 10^{-2}$	$6.73 \times 10^{-4}$	0.0380	$4.37 \times 10^{-2}$	$2.07 \times 10^{-3}$	0.1060	0.0725	0.3901	0.27
Tsitsiklis	$2.14 \times 10^{-2}$	$4.89 \times 10^{-4}$	0.0281	$3.81 \times 10^{-2}$	$1.57 \times 10^{-3}$	0.0825	0.1007	0.4348	0.26
MSFM	$2.36 \times 10^{-2}$	$6.07 \times 10^{-4}$	0.0349	$4.23 \times 10^{-2}$	$1.94 \times 10^{-3}$	0.1007	0.0825	0.3572	0.29
SGFM	$2.33 \times 10^{-3}$	$6.32 \times 10^{-6}$	0.0051	$1.25 \times 10^{-2}$	$2.14 \times 10^{-4}$	0.0580	0.0022	0.0277	0.33
Linear4	$1.10 \times 10^{-2}$	$1.71 \times 10^{-4}$	0.0285	$1.69 \times 10^{-2}$	$4.01 \times 10^{-4}$	0.0875	0.0122	0.1036	0.51
Linear8	$2.25 \times 10^{-3}$	$6.82 \times 10^{-6}$	0.0046	$4.46 \times 10^{-3}$	$3.43 \times 10^{-5}$	0.0596	0.0028	0.0355	0.52
IsoLinear	$2.25 \times 10^{-3}$	$6.82 \times 10^{-6}$	0.0046	$4.03 \times 10^{-3}$	$3.11 \times 10^{-5}$	0.0596	0.0109	0.0911	0.91
Bilinear8	$2.74 \times 10^{-3}$	$9.42 \times 10^{-6}$	0.0052	$5.01 \times 10^{-3}$	$4.10 \times 10^{-5}$	0.0607	0.0028	0.0101	0.65
Up4	$1.79 \times 10^{-3}$	$7.60 \times 10^{-6}$	0.0101	$3.14 \times 10^{-3}$	$2.89 \times 10^{-5}$	0.0655	0.0451	0.1919	1.37
Up8	$2.99 \times 10^{-4}$	$1.96 \times 10^{-7}$	0.0014	$1.54 \times 10^{-3}$	$7.81 \times 10^{-6}$	0.0289	0.0011	0.0221	1.42
UpSG	$1.96 \times 10^{-3}$	$4.15 \times 10^{-6}$	0.0035	$1.20 \times 10^{-2}$	$1.94 \times 10^{-4}$	0.0566	0.0015	0.0141	1.42

We also enlist the computation times for each of these methods on a 500x500 grid in the last column of Table 2. All computation times were measured on a laptop with a 1.73 GHz Processor.

## 5 Conclusion

In this paper we present techniques to make the fast marching method independent of the marching direction and thus improve the accuracy of the Fast Marching Method. One approach interpolates the local traveling cost along the front and the other computes  $u$  on an upsampled grid. We also showed that combining the 8 and 4-connected neighbor schemes further reduces the inaccuracy by considering all possible directions of the arrival of the front. We have compared both our approaches to the existing Fast Marching techniques and we have shown a significant improvement over them. Although both our approaches have higher computation times, they can be implemented efficiently on hardware and they are practical solutions to eliminate the inaccuracies of existing techniques.

## Acknowledgements

This work was partially supported by NIH grant R01-HL-085417, NSF grant CCF-0728911 as well as an EmTech seed grant.

## References

1. Adalsteinsson, D., Sethian, J.A.: A fast level set method for propagating interfaces. *Journal of Computational Physics* 118, 269–277 (1994)
2. Bronstein, A.M., Bronstein, M.M., Devir, Y.S., Kimmel, R., Weber, O.: Parallel algorithms for approximation of distance maps on parametric surfaces (2007)

3. Chan, T., Vese, L.: An active contour model without edges. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) *Scale-Space 1999*. LNCS, vol. 1682, pp. 141–151. Springer, Heidelberg (1999)
4. Cohen, L., Kimmel, R.: Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision* 24, 57–78 (1997)
5. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: A minimal path approach. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 666 (1996)
6. Danielsson, P.E., Lin, Q.: A modified fast marching method. In: Bigun, J., Gustavsson, T. (eds.) *SCIA 2003*. LNCS, vol. 2749, pp. 1154–1161. Springer, Heidelberg (2003)
7. Hassouna, M.S., Farag, A.A.: Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(9), 1563–1574 (2007)
8. Kim, S., Folie, D.: The group marching method: An  $o(n)$  level set eikonal solver
9. Polymenakos, L.C., Bertsekas, D.P., Tsitsiklis, J.N.: Implementation of efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 43, 278–283 (1998)
10. Sethian, J.A.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge (1999)
11. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions On Automatic Control* 40(9), 1528–1538 (1995)

# Descattering Transmission via Angular Filtering

Jaewon Kim<sup>1,2</sup>, Douglas Lanman<sup>1,\*</sup>,  
Yasuhiro Mukaigawa<sup>1,\*\*</sup>, and Ramesh Raskar<sup>1</sup>

<sup>1</sup> MIT Media Lab

<sup>2</sup> Korea Institute of Science and Technology(KIST)

**Abstract.** We describe a single-shot method to differentiate unscattered and scattered components of light transmission through a heterogeneous translucent material. Directly-transmitted components travel in a straight line from the light source, while scattered components originate from multiple scattering centers in the volume. Computer vision methods deal with participating media via 2D contrast enhancing software techniques. On the other hand, optics techniques treat scattering as noise and use elaborate methods to reduce the scattering or its impact on the direct unscattered component. We observe the scattered component on its own provides useful information because the angular variation is low frequency. We propose a method to strategically capture angularly varying scattered light and compute the unscattered direct component. We capture the scattering from a single light source via a lenslet array placed close to the image plane. As an application, we demonstrate enhanced tomographic reconstruction of scattering objects using estimated direct transmission images.

**Keywords:** computational photography, direct transmission, scattered transmission, multiple scattering, image decomposition.

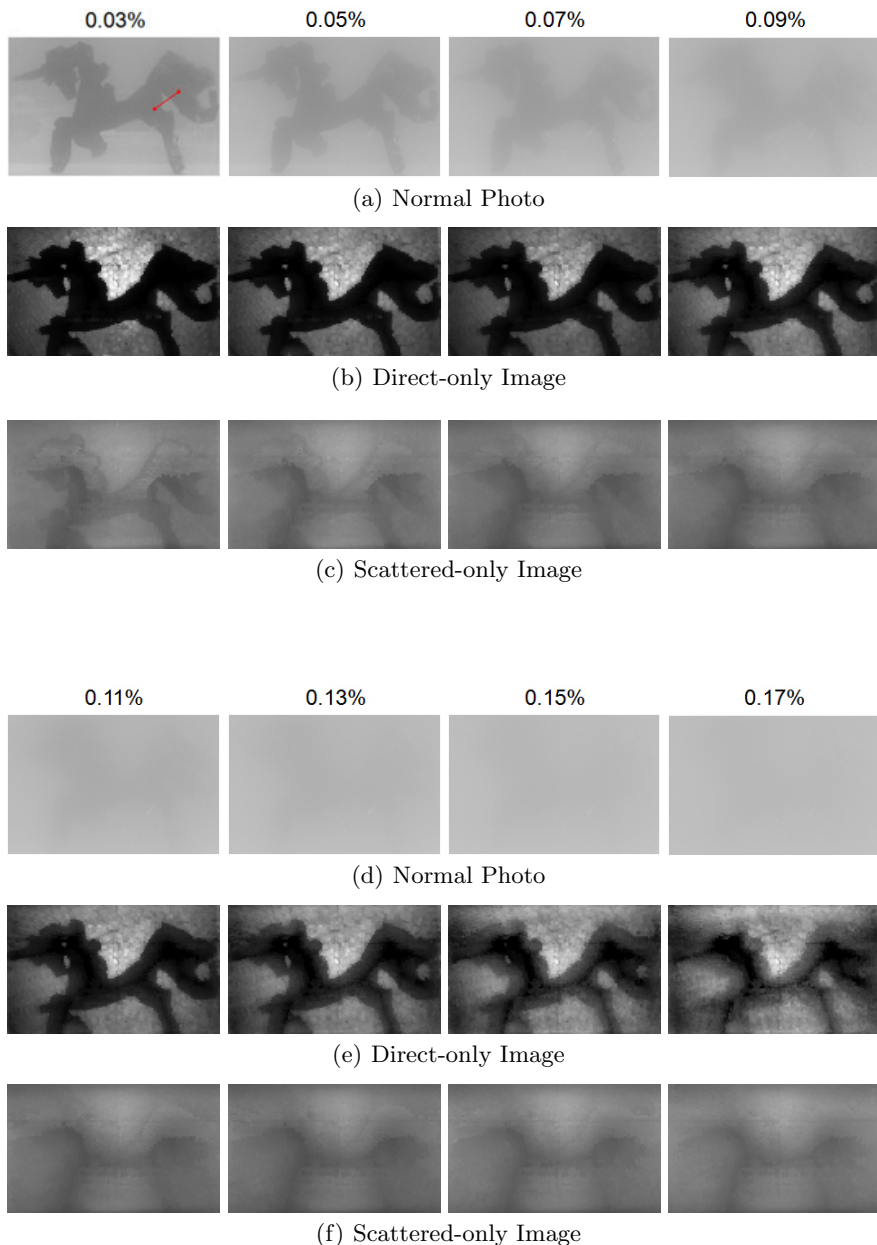
## 1 Introduction

The separation of direct and scattered components of incident light is a challenging topic in computer vision and graphics, a task that is confounded by the complex behavior of light in participating media, e.g., reflection, refraction, and scattering in haze, underwater or in volumetric translucent objects. These complex characteristics of light are one of the main factors hindering an analytical solution for direct-scattered separation. For this reason, active coding methods have been proposed. Nayar et al. [1] project high-frequency patterns onto a reflective scene. Such active coding methods achieve accurate and robust separation. Narasimhan et al. [2] use structured light to estimate the 3-D shape of objects in scattering media, including diluted suspensions. Gu et al. [3] also use structured light, exploiting compressive sensing techniques, to decrease the data acquisition time. Atcheson et al. [4] estimate the 3-D shape of non-stationary

---

\* Visiting from Brown University.

\*\* Visiting from Osaka University.



**Fig. 1.** Recovery of an opaque object in participating media with milky water. (a) and (d) Normal photos according to concentration 0.03%–0.17% in which water is 7500ml and milk is increased by 1.5ml from 2ml. (b) and (e) Recovered direct-only images computed using angular-domain filtering with a lenslet array. Note enhanced visibility for sharp features and edges in the descattered image. (c) and (f) scattered-only images preprocessed to acquire the direct-only images.

gas flows. In many existing approaches, only scattering scenes composed of low density materials (eg. smoke, liquid, and powder) are allowed, such that a single scattering mode is dominant. Using the methods outlined in this paper, we demonstrate direct-scattered separation for scenes in which multiple scattering is predominant.

In this paper we use a passive, single-shot imaging method to achieve separation of *transmitted* light for a scene containing heterogeneous scattering media. Specifically, we use a lenslet (or pinhole) array close to the image plane to separate direct and scattered components of incident light (albeit while reducing the resolution of the recovered direct and scattering component images since the projection of each lenslet or pinhole provides a single pixel in the recovered images). Using a sequence of such images, we are able to recover an estimate of the volumetric attenuation using existing tomographic reconstruction methods, demonstrating benefits for both dehazing and 3D shape recovery.

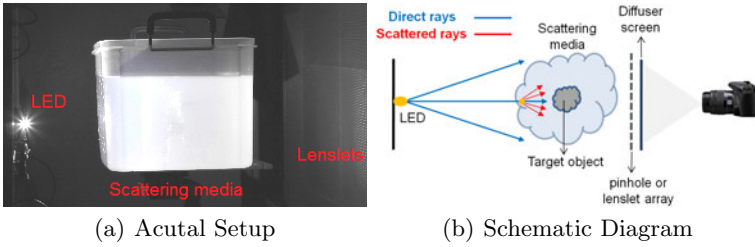
## 1.1 Contributions

We describe a method for single-exposure separation of direct and scattered components of transmitted light passing through scattering media using a lenslet or pinhole array placed closely to the image sensor. In the direct-only image, high-frequency details are restored and provide strong edge cues for scattering objects. Due to its single-shot nature, this method is well-suited for analyzing dynamic scenes. We demonstrate enhanced tomographic reconstruction of scattering objects using direct component images. These separation methods are well-suited for applications in medical imaging, providing an internal view of scattering objects such as human skin using visible or near-visible wavelength light sources, rather than X-rays.

## 1.2 Related Work

**Direct-Scattered Separation:** Direct-scattered separation of light is widely studied in diverse fields spanning computer vision, graphics, optics, and physics. Due to the complexities of scattering, reflection, and refraction, analytical methods do not achieve satisfactory results in practical situations. In computer vision and graphics, Nayar et al. [1] present an effective method to separate direct and scattered components from a scene by projecting a sequence of high-frequency patterns. Their work is one of the first to handle arbitrary natural scenes. However, it requires temporally-multiplexed illumination, limiting the utility for dynamic scenes. Nasu et al. [5] present an accelerated method using a sequence of three patterns. In addition, Rosen and Abookasis [6] present a descattering method using speckle analysis.

**Microscopy:** The scattering in microscopic objects is addressed by careful optical methods. Hisashi [7] present a method to achieve a sharp in-focus signal in a confocal microscope setup. They use two pinholes to sense in- and out-of-focus signals and acquire a sharp in-focus signal by subtracting the two. This requires two exposures and careful alignment for each spot. In addition, scanning process



**Fig. 2.** (a) Imaging system consisting of an LED and a lenslet array. A single LED is used to back-illuminate a scattering scene. A diffuser is used to form an image through a lenslet array. A high-resolution camera captures the array of lenslet images in a single exposure. (b) A schematic diagram of the actual setup.

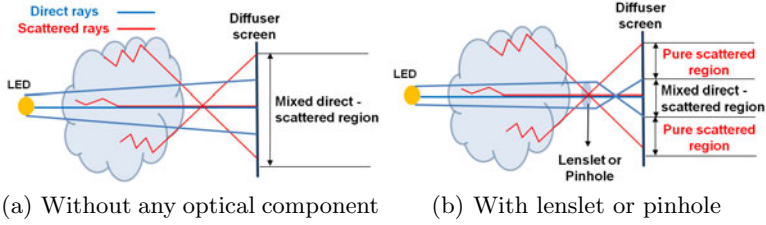
is required to recover a whole scene. Sheppard et al. [8] present a method to improve lateral and axial resolution. They achieve enhanced lateral resolution by subtracting a weighted traditional signal from a confocal imaging signal. Their method also increases axial resolution by using a number of detectors in different sizes. These are multi-exposure methods but are similar to our method where a simple linear transformation of intensities in a neighborhood recovers the sharp component. Levoy et al. [9] record 4D light field using a microlens array for digital refocusing and acquiring angular views with a single snapshot photo in microscope. In our method, the angular variation recorded by the similar way is exploited explicitly making it robust to non-homogeneous local variations. Our method requires no complicated light sources or mechanical scanning or change in aperture settings.

**3D Recovery in Scattering Media:** Narasimhan et al. [2] and Gu et al. [3] use sequential structured light patterns to recover 3D shape of static opaque objects in low density scattering media. Our method requires simple light sources and only a single photo per view. Atcheson et al. [4] recover non-stationary gas flows using Schlieren imaging and multiple cameras. The method is suitable for refracting but not scattering media. Rosen and Abookasis [6] proposed a method to recovery shape of binary objects between 2 layers of scattering media based on refocusing principles. Trifonov et al. [10] consider tomographic reconstruction of transparent objects using large number of photos and index matching liquids. Our emphasis is on scattering objects.

## 2 Imaging System

### 2.1 Overview

The actual setup and schematic diagram of our proposed imaging system is shown in Figure 2. We note that our direct-scattered separation method handles solid objects and liquid mixtures. In particular, we consider the case when a solid object is enclosed by a scattering media, as is typical in medical imaging

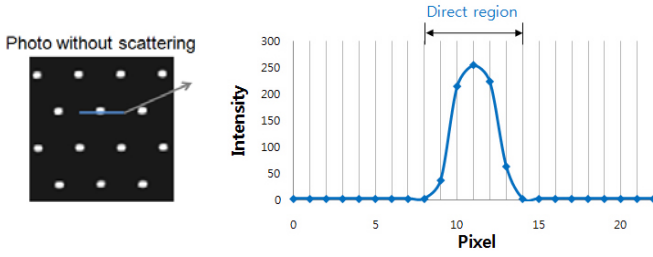


**Fig. 3.** Recovering the direct component from a mixed direct-scattered region. There is no way to separate direct and scattered rays in (a). The rays are spatially separated by a lenslet or pinhole in (b). We use the estimate from the pure-scattered region to subtract the scattered component in the central region.

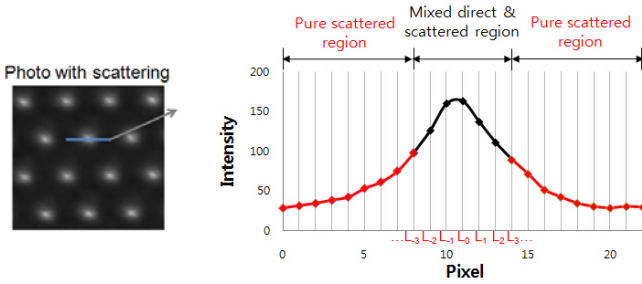
applications. The property of the solid object can be most types of scattering except significant internal refraction. Refraction is treated as scattering and appropriately separated from direct component but the low frequency fitting of our method(RTE scattering model) becomes inaccurate. Thin glass objects and thin boundary of media with minor refraction are fine. Under a simple geometric scattering model, light rays are emitted from a point source (an LED in our system). When each direct ray impinges on a scattering center, a new scattering light source is effectively created(Figure 2(b)). Both direct and scattered rays form an image through a pinhole or lenslet array onto a diffuser screen which is captured by a camera. We apply radiative transport equation(RTE) [11] to model the angular variation of this scattering center. We assume, at a basic level, that the heterogenous scattering media will be dominated by multiple scattering events [12] [13].

## 2.2 Imaging with Lenslets or Pinhole Arrays

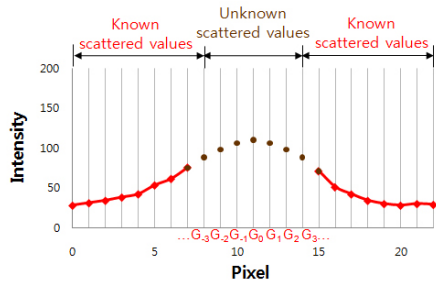
We use a Canon EOS Digital Rebel XSi, with a resolution of  $4272 \times 2848$  pixels. The lenslet array is separated from the diffuser in order to form an image of the scattering volume focusing on the entire volume with large DOF (centimeters almost infinite) of lenslet(Figure 2(b)). Lanman et al. [14] used a similar imaging setup to compute a single shot lightfield of opaque objects while we address scattering to compute direct component of translucent objects. From Figure 3(b), we infer there are two regions in the image under each lenslet. The first region consists of a mixed signal due to cross-talk between the direct and scattered components. The second region represents a pure scattered component. In the following section, we show a simple method for analyzing such imagery to separate direct and scattered components for multiple-scattering media. As shown in Figure 3(b), the angular sample directly under each lenslet can be used to estimate the combined direct plus scattered transmission along the ray between a given pixel and the light source. Similarly, any non-zero neighboring pixels(not beneath the lenslet) can be fully attributed to scattered illumination due to volumetric scattering.



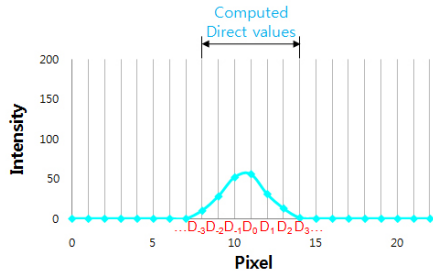
(a) Without Scattering



(b) With Scattering



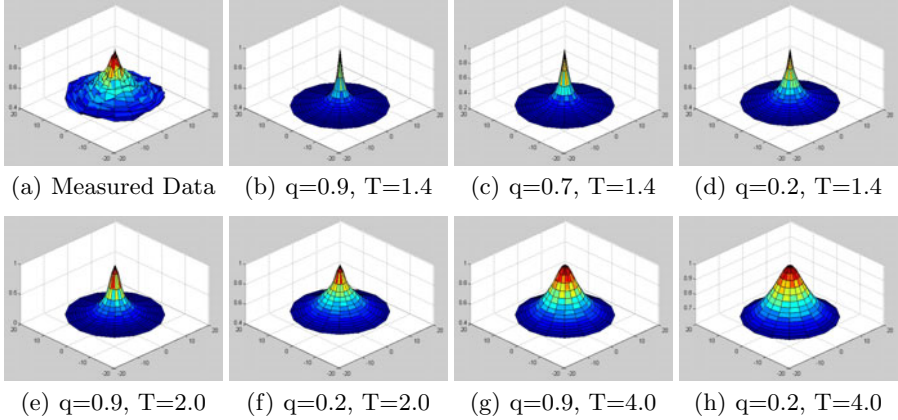
(c) Estimation



(d) Subtraction

**Fig. 4.** Comparison of direct and scatter components without and with scattering media. (a) Central region under each lenslet is sharp without scattering. (b) Direct as well as scattered component is included in the central region. (c) Measured (red) and estimated (brown) values for scattering-only component. (d) The direct-only image formed by subtracting (c) from (b).





**Fig. 5.** RTE(Radiative Transport Equation) modeling of scattering values through a pinhole. (a) Measured data (b)-(h) RTE Modeling with different  $q$  and  $T$ . (f) is in minimum fitting error with the measured data.

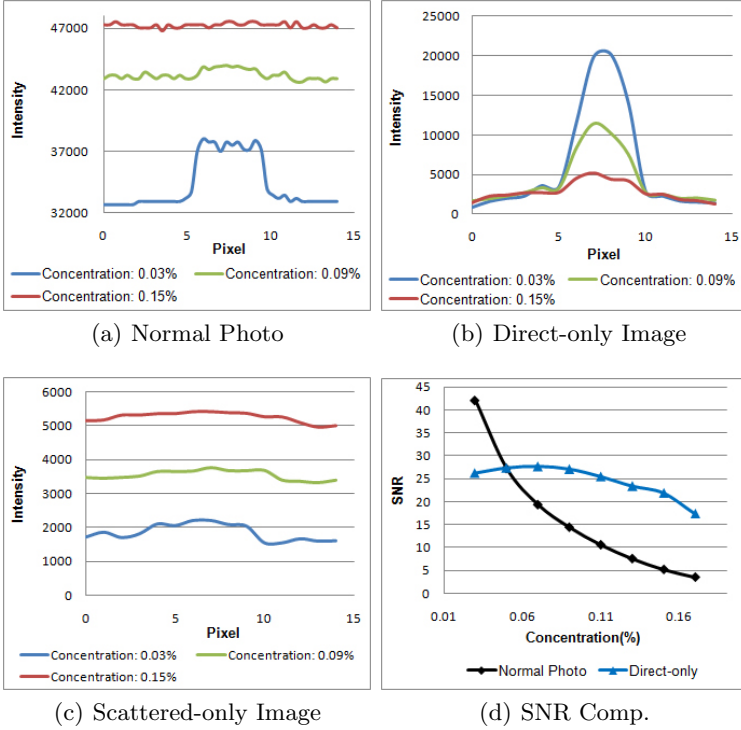
### 3 Direct-Scattered Separation

#### 3.1 Separation via Angular Filtering

In this section we consider direct-scattered separation for a 1-D sensor and a 2-D scene, while the results can be trivially extended to 2-D sensors and 3-D volumes. In the following analysis, we consider only lenslet arrays, however a similar analysis holds for pinhole arrays. As shown in Figure 4(a), the diffuser-plane image, a reference image to be captured in a calibrated setup, consists of a set of sharp peaks under each lenslet in the absence of any scattering media between the light source and diffuser. As shown on 4(b), the lenslet images contain extended, blurred patterns when a scattering object is placed between the light source and camera. Ultimately, the scattered light causes samples to appear in pixels neighboring the central pixel under each lenslet. A single lenslet image is defined by two separate regions: a pure scattered component region and a region of mixed direct and scattered components. We represent the received intensity at each diffuser-plane pixel as,  $\{L_0, L_1, \dots, L_n\}$ , when a scattering object is placed between the light source and the diffuser. The individual sensor values are modeled as

$$\begin{aligned}
 L_0 &= G_0 + D_0 \\
 &\vdots \\
 L_n &= G_n + D_n,
 \end{aligned} \tag{1}$$

where  $\{G_n\}$  and  $\{D_n\}$  represent the underlying scattered and direct intensities measured in the sensor plane, respectively. As shown in Figure 4(b), a straightforward algorithm can be used to estimate the direct and scattered components

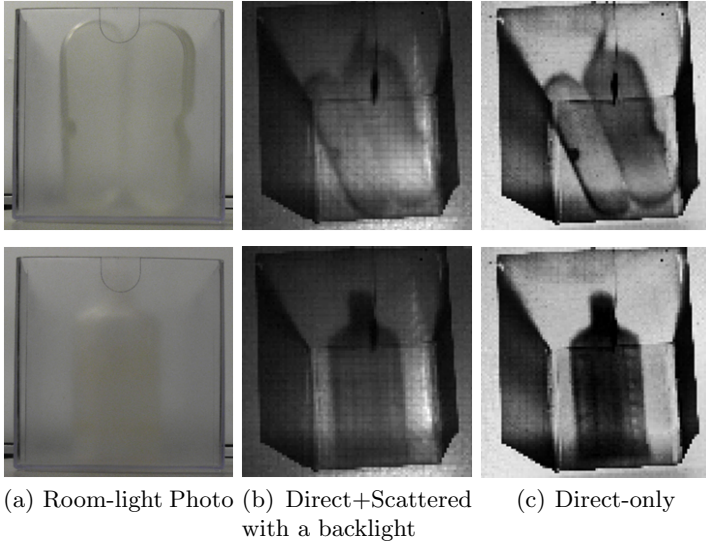


**Fig. 6.** (a)-(c) Intensity profiles show that signals in direct-only images are significantly enhanced compared with those in normal photos (Intensity scale 0-65535) (d) SNR comparison between normal and direct-only images shows that our method is effective at scattering-dominant scene.

received at each lenslet. First, we estimate the non-zero region in (a) which is captured with no object present. Next, we approximate values of the scattering component  $\{G_n\}$  in the region using a scattering model, described in next section, as shown in (c). Note that this region is subject to mixing in (b) and the scattering component must be approximated from the known scattered values in (c). Finally, a direct-only image can be estimated by subtracting the estimated scattering component for the central pixel under a lenslet, such that  $D_0 \approx L_0 - G_0$ .

### 3.2 Mathematical Model for Multiple Scattering

We describe the multiple scattering model used in the descattering algorithm described in the previous section. Numerical Monte-Carlo techniques have been widely used for tracing scattered rays but it needs high computational cost for a large number of rays. To implement efficient descattering algorithm, we use the physics-based model presented by Narasimhan and Nayar [11]. Multiple scattered intensity through a pinhole can be described by RTE (Radiative Transport



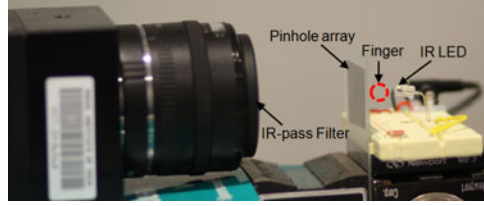
**Fig. 7.** Direct-scattered separation with a pinhole array. Direct-only images enhance high frequency features of an object enclosed by a scattering solid object.

Equation) and the solution of it is a function of three parameters,  $T$ (optical thickness),  $q$ (forward scattering parameter) and  $x$ (spatial position) as explained in the Narasimhan and Nayar’s paper. RTE is used to fit measured 2D data, Figure 5(a), under each lenslet of our imaging condition. (b)-(h) show varied intensity distributions according to different  $T$  and  $q$ . By an iterative error-minimization method, the best matching profile, (f), can be found for the measured 2D signal (a) and any unknown scattered value for nearby regions can be approximately calculated by the fitted model.

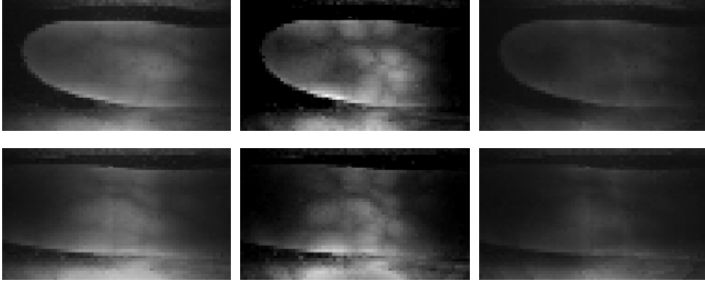
### 3.3 Experimental Results

From Section 3.1, we separate direct signals  $\{D_n\}$  and scattered signals  $\{G_n\}$  in each lenslet region. By collecting and combining the direct signals in each lenslet (or pinhole) region, we can generate a direct image. The scattered image is obtained by a similar process, collecting scattered signals. The original image(or normal photo) can be considered as the summed image of the direct and scattered signals. The resolution of the direct and scattered component images are identical to the number of lenslets (or pinholes), because there is only one signal value for direct and scattered components for each lenslet region. In our experiment, the image size is  $150 \times 100$ .

Figure 4 compare normal photos of a scattering scene, consisting of an opaque horse-shape object enclosed in an aquarium with milky water, and direct-only images generated by our proposed separation process in lenslet array setup. From left to right, the images show results acquired at higher concentrations of milky water. Figure 6 (a)-(c) compare signals at the position of the red line



(a) IR Imaging Setup



(b) Normal Photo

(c) Direct-only

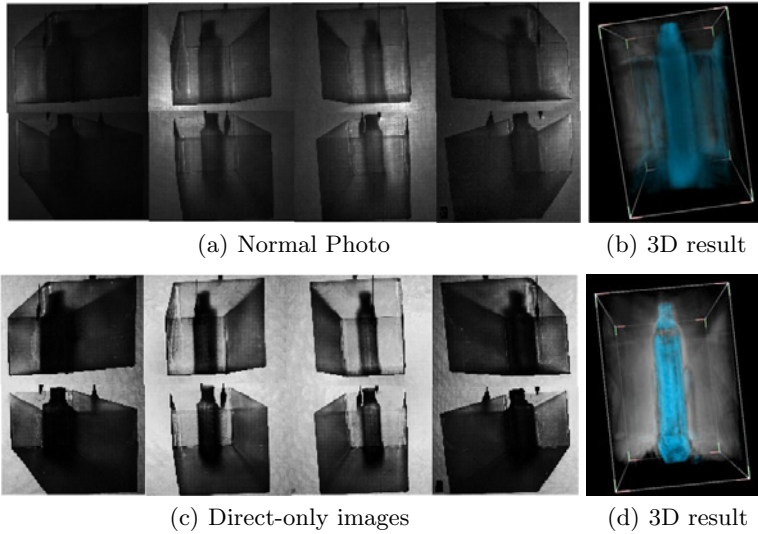
(d) Scattered-only

**Fig. 8.** Direct-scattered separation images for human fingers using infrared imaging setup. Direct-only images show sharper shapes of finger veins than normal photos. (a) The camera focuses on the pinhole array plane. The IR-pass filter cut visible light and only pass IR light.

in Figure 7(a) for normal photos, direct-only images and scattered only images at different concentrations. (b) shows the signals are enhanced compared with signals in normal photos, (a). As the concentration of milky water is increased, the intensity of the signal in direct-only images, (b), is decreased. The opposite effect is observed in scattered-only images, (c), which follows physical reasoning. (d) compares the signal-to-noise ratio(SNR) between normal photos and direct-only images according to concentration. At low concentration, the SNR of a normal photo is larger than one of a direct-only image. However, as concentration is increased, the SNR of a direct-only image gradually becomes higher than the SNR of a normal photo. Note that the signal of normal photos, (a), is significantly decreased from the concentration 0.03% to 0.09% compared with the signal change in direct-only images in (b).

Figure 7 shows experimental results using a pinhole array instead of a lenslet array. (a) shows the room-light photo of a solid object placed in the scattering medium. (b) displays ground-truth photos which are acquired by summing all direct and scattered values under each lenslet. (c) contains the direct-only image. By comparing (b) and (c), we find that the direct-only images give the sharpest image boundaries for the scattering objects.

We tested our method for human fingers with a near-infrared imaging setup where finger veins are well visualized with infrared light. Direct-only images in Figure 8 (c) shows sharper shape of the veins than normal photos do. This is an initial experiment for a strongly scattering challenging object but the image



**Fig. 9.** Tomographic reconstruction results. (a) and (b) show eight normal photos captured at different light-projection angles and 3D result using them, respectively. (c) and (d) are direct-only images for the each normal photo and 3D result using the direct-only images.

formation model is identical to Figure 11 and our results are comparable to [15] which uses specialized narrow wavelength profile light source. As in Figure 11, veins closer to the skin are more prominently visible as they are decomposed in the direct component although the finger is strongly scattering. Such enhanced visualization of a human body will benefit medial, biometrics and HCI applications.

### 3.4 Volumetric Reconstruction Using ART

We use an algebraic reconstruction technique (ART) presented by Roh et al. [16] to reconstruct 3-D shape of scattering objects following traditional short-baseline tomography approaches. Figure 9(b) and (d) compare two 3-D reconstruction results using eight normal photos and eight descattered images. We captured eight photos sequentially with a LED mounted at different position to get multiple angular views of the targeted inside object, the bottle, in Figure 7 (bottom). In the 3D reconstruction, Figure 9(b), the bottle is rendered by blue color to add distinguishability from the outer scattering container. Note that the rendering isn't accurate for the bottle since the bottle shape in the captured images has been hazed by scattering. The 3D result using direct-only images in (d) shows a more accurate rendering result for the inside bottle object.

## 4 Benefits and Limitations

**Benefits:** This paper makes three primary contributions: (1) robust separation of direct and scattered components of incident light passing through heterogenous

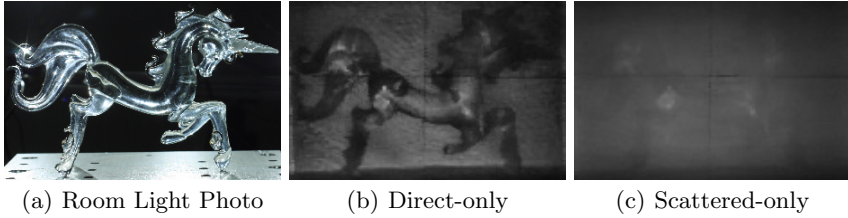


**Fig. 10.** Direct-only images by Nayar et al. [1] method for a horse-shaped object enclosed in an aquarium with diluted milk(Concentration 0.11%) (a) Inset of a captured photo with a projected high-frequency pattern (b) Direct-only image with wide projector’s DOF (c) Direct-only image with short projector’s DOF

scattering media, (2) 3-D volumetric reconstruction of the mixed scattering objects, and (3) a novel technique to enable effective volumetric analysis of solid scattering objects in multiple-scattering conditions. For direct and scattered separation, our method requires only a simple system consisting of a lenslet(or pin-hole array) and a point light source. Compared with other methods, like using a projector to generate temporally-multiplexed patterns, our method can achieve separation in a single exposure. Also, our method requires simple local computations, performed independently on each lenslet image. Furthermore, dynamic applications are possible.

Our 3-D reconstruction technique for scattering objects has potential applications extending beyond computer vision and graphics, including non-invasive medical imaging [17]. Specific parts of biological organisms, including human fingers, can be modeled by scattering and translucent material similar to objects considered in this work. As a result, it is possible to view the inner 3-D shape of certain parts in human and animal bodies by this technique. Most importantly, such visible-wavelength separation methods may allow hazardous X-ray imaging to be replaced in certain applications. Such applications include the personal identification field. For example, 3-D shape recognition of finger veins can provide strong cues for identification. Furthermore, such features may overcome several limitations of traditional fingerprints, which change due to aging.

For transmission-mode descattering, the proposed method has several unique advantages in comparison to the closely related method of Nayar et al. [1]. One of the key limitations of Nayar et al. [1] is that the assumption of high-frequency projected patterns aren’t satisfied in dense media(Figure 10(a)). Another limitation of any projector-based solution, such as that of Nayar et al., arises due to the finite DOF(Depth of Field) achieved in practice. For transmission-mode descattering, the projector must focus on the scattering media and the screen at the same time—unlike the case of reflection-mode acquisition. Thus, the projector requires a wide DOF. Figure 10(c) shows a direct image by [1] when the projector’s DOF isn’t wide enough to cover both inside object and screen. Our proposed method is free from such focusing problems. Furthermore, our proposed



**Fig. 11.** Limitation of refraction. Our method results for a translucent horse-shaped object enclosed in an aquarium with diluted milk(Concentration 0.11%).

imaging system is much simpler and inexpensive, containing a single LED. Finally, our method is well-suited for less dense parts of human bodis, such as fingers as shown in Figure 8.

**Limitations:** In the current method, the primary limitation is due to the loss of resolution incurred by the lenslet or pinhole array. In addition, pinhole arrays require long exposures. While lenslets could be used to overcome exposure issues, loss of resolution remains. Also, the separated results can be affected by refraction. Figure 11 shows separated results of a translucent horse-shape object in milky water. Note that the legs and the end of the tail in (b) look dark by refraction although they have similar density with the body area as shown in (a). The proposed 3D reconstruction method requires control of the lighting environment and, as a result, cannot be directly extended to natural environments. Furthermore, this reconstruction method requires a temporally-multiplexed set of images for tomographic reconstruction, limiting dynamic scene reconstruction. We emphasize, however, that direct-scattered separation can be performed in a single exposure. Most importantly, we anticipate challenges in strongly-scattering environments. In such circumstances, the scattering term will dominate the proposed low-order polynomial approximation and the direct term will not be reliably recovered.

## 5 Conclusion

In this paper, we show a new method to separate direct and scattered components of transmitted light from translucent objects. The direct-only images provide sharp shape information for such scattering objects. We have demonstrated a volumetric reconstruction technique, following classic methods of limited-baseline tomography, to reconstruct scenes using direct-only images. These results can be achieved with low-cost hardware consisting of LEDs, diffusers, and lenslet array(or printed pinhole array mask). In particular, we show that visible-wavelength radiation can be applied for attenuation-based tomography when such separation methods exist in the transmission-mode. We hope that our research will inspire others to pursue low-energy, non-invasive imaging in the medical and biological sciences.

## Acknowledgments

Ramesh Raskar is supported by an Alfred P. Sloan Research Fellowship.

## References

1. Nayar, S., Krichnan, G., Grossberg, M., Raskar, R.: Fast separation of direct and global components of a scene using high frequency illumination. In: ACM TOG, vol. 12, pp. 935–943 (2006)
2. Narasimhan, S.G., Nayar, S.K., Sun, B., Koppal, S.J.: Structured light in scattering media. In: Proc. IEEE ICCV, vol. 1, pp. 420–427 (2005)
3. Gu, J., Nayar, S., Grinspun, E., Belhumeur, P., Ramamoorthi, R.: Compressive structured light for recovering inhomogeneous participating media. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 845–858. Springer, Heidelberg (2008)
4. Atcheson, B., Ihrke, I., Heidrich, W., Tevs, A., Bradley, D., Magnor, M., Seidel, H.P.: Time-resolved 3D capture of non-stationary gas flows. In: ACM TOG (2008)
5. Nasu, O., Hiura, S., Sato, K.: Analysis of light transport based on the separation of direct and indirect components. In: PROCAMS (2007)
6. Rosen, J., Abookasis, D.: Noninvasive optical imaging by speckle ensemble. *Optics Letters* 29 (2004)
7. Okugawa, H.: A new imaging method for confocal microscopy. In: SPIE (2008)
8. Sheppard, C.J.R., Cogswell, C.J.: Confocal microscopy with detector arrays. *Journal of Modern Optics* 37, 267–279 (1990)
9. Levoy, M., Zhang, Z., McDowall, I.: Recording and controlling the 4d light field in a microscope using microlens arrays. *J. of Microscopy* (2009)
10. Trifonov, B., Bradley, D., Heidrich, W.: Tomographic reconstruction of transparent objects. In: Eurographics Symposium on Rendering (2006)
11. Narasimhan, S.G., Nayar, S.K.: Shedding light on the weather. In: Proc. IEEE CVPR, vol. 151, pp. 665–672 (2003)
12. Jensen, H., Marschner, S., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. In: SIGGRAPH, pp. 511–518 (2001)
13. Sun, B., Ramamoorthi, R., Narasimhan, S.G., Nayar, S.K.: A practical analytic single scattering model for real time rendering. In: TOG, pp. 1040–1049 (2005)
14. Lanman, D., Raskar, R., Agrawal, A., Taubin, G.: Shield fields: Modeling and capturing 3d occluders. In: SIGGRAPH Asia 2008 (2008)
15. Miura, N., Nagasaka, A.N., Miyatake, T.: Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Machine Vision and Applications* (2004)
16. Roh, Y.J., Park, W.S., Cho, H.S., Jeon, H.J.: Implementation of uniform and simultaneous ART for 3-D reconstruction in an x-ray imaging system. In: IEEE Proceedings, Vision, Image and Signal Processing, vol. 151 (2004)
17. Tuchin, V.: *Tissue Optics: Light Scattering Methods and Instruments for Medical Diagnosis*. SPIE Publications (2007)



# Flexible Voxels for Motion-Aware Videography

Mohit Gupta<sup>1</sup>, Amit Agrawal<sup>2</sup>,  
Ashok Veeraraghavan<sup>2</sup>, and Srinivasa G. Narasimhan<sup>1</sup>

<sup>1</sup> Robotics Institute, Carnegie Mellon University, Pittsburgh, USA

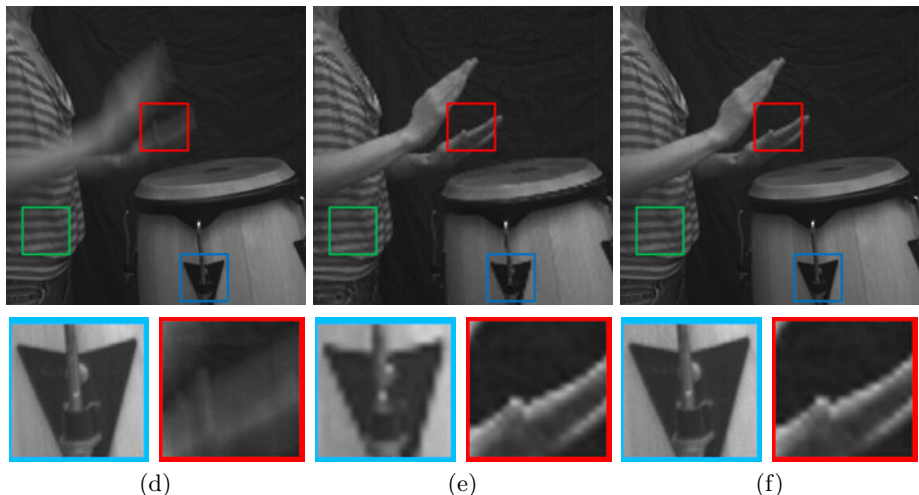
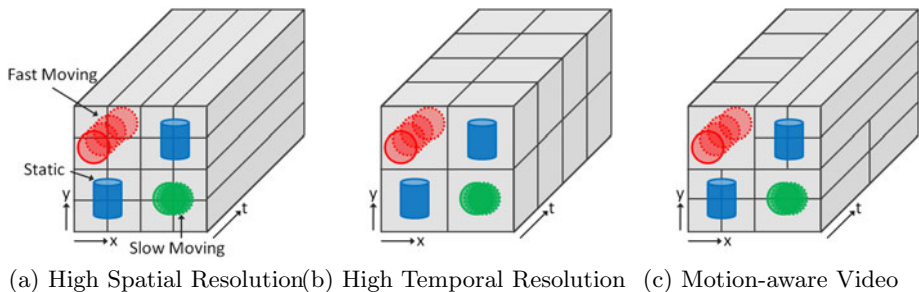
<sup>2</sup> Mitsubishi Electrical Research Labs, Cambridge, USA

**Abstract.** The goal of this work is to build video cameras whose spatial and temporal resolutions can be changed post-capture depending on the scene. Building such cameras is difficult due to two reasons. First, current video cameras allow the same spatial resolution and frame rate for the entire captured spatio-temporal volume. Second, both these parameters are fixed *before* the scene is captured. We propose different components of video camera design: a sampling scheme, processing of captured data and hardware that offer post-capture variable spatial and temporal resolutions, independently at each image location. Using the motion information in the captured data, the correct resolution for each location is decided automatically. Our techniques make it possible to capture fast moving objects without motion blur, while simultaneously preserving high-spatial resolution for static scene parts within the same video sequence. Our sampling scheme requires a fast per-pixel shutter on the sensor-array, which we have implemented using a co-located camera-projector system.

## 1 Introduction

Traditional video cameras offer a fixed spatial resolution (SR) and temporal resolution (TR) independent of the scene. Given a fixed number of measurements (voxels) to sample a space-time volume, the shape of the voxels can vary from ‘thin and long’ (high SR, low TR) to ‘fat and short’ (high TR, low SR) as shown in Figure 1. For conventional cameras, the shape of the voxels is fixed *before* capture (scene independent), and is the same for the entire spatio-temporal volume. Can we design video cameras that can choose different spatio-temporal resolutions post-capture, depending on the scene content? We show that it is achievable by a careful choice of per-pixel temporal modulation along with well-designed reconstruction algorithms.

While a high spatial resolution camera captures the fine detail in the static scene parts, it blurs fast moving objects. On the other hand, a high-speed camera captures fast temporal variations but unnecessarily trades off light throughput and spatial resolution for the static and slowly moving scene parts. This fundamental capture limitation can be overcome by designing video cameras with the following two properties: (a) The flexibility to decide the spatio-temporal resolution *post-capture* in a content-aware (scene dependent) manner, and (b)



**Fig. 1. Different samplings of the space-time volume:** For conventional video cameras, the sampling of the space-time volume is decided *before* the scene is captured. Given a fixed voxel budget, a high spatial resolution (SR) camera (a) results in large motion blur and (d) aliasing. A high-speed camera (b) results in low SR even for the static/slow-moving parts of the scene (drums in (e)). With our sampling and reconstruction scheme, the spatio-temporal resolution can be decided *post-capture*, *independently at each location* in a content-aware manner (c): notice the reduced motion blur for the hands (f) and high SR for the slow-moving parts of the scene.

the ability to make this choice *independently* at each video location. In this paper, we take an initial step towards achieving these goals by demonstrating a hardware setup that enables fast per-pixel temporal modulation, by designing a necessary space-time sampling scheme and by developing simple yet effective motion-aware post-processing interpolation schemes.

We determine necessary conditions for a sampling scheme to allow capturing multiple space-time resolutions simultaneously. Data captured with a sampling scheme which satisfies these conditions can be reconstructed at different spatio-temporal resolutions, independently at each image location. The reconstruction problem is posed as interpolation of scattered samples using well-known anisotropic diffusion techniques. Since the shape of diffusion tensor determines

the local smoothing orientations, by designing different diffusion tensors, we can essentially achieve a *continuum* of effective spatio-temporal resolutions. The *correct* resolution is automatically determined by designing spatially and temporally varying local diffusion tensors based on motion information in the captured data.

Hardware implementation of our sampling scheme requires fast *independent* shutter control of each pixel, which is not possible with available commercial cameras. We have built a prototype using a projector-camera setup which achieves rapid per-pixel temporal modulation during camera integration time. This setup emulates a flexible spatio-temporal resolution camera with a maximum frame rate of 240 Hz, even though the frame rate of the original camera is only 15 Hz. We show several real results that demonstrate variable resolution trade-off in space and time post capture.

## 1.1 Related Work

**Content-based re-sampling and compressive sampling:** Content-based re-sampling and representation of data is central to most image/video compression algorithms. Adaptive sampling of data has been used for building content-aware multi-resolution image and video pyramids for fast data transmission [1]. Recently, the field of compressive sensing has exploited sparsity in data at acquisition time, thus reducing the sensing over-head significantly [2,3]. In contrast, our sampling scheme allows re-allocating the saved resources to another dimension in a content-aware manner. If the captured video-stream is sparse in spatial domain, high-frequency detail can be preserved in the temporal dimension and vice-versa.

**Multi-dimensional imaging:** Several methods trade off spatial resolution to sample other dimensions such as dynamic range [4], wavelength [5], angular dimensions in lightfield [6] and color/polarization [7]. Ben-Ezra et al. [8] used precise sub-pixel detector shifts for increasing the spatial resolution of a video camera. In contrast, our goal is to increase TR much beyond the native frame rate of the camera by trading off SR. Recently, a variety of approaches [9,10,11] which increase TR by trading off SR have been introduced. However, these methods provide *the same* spatio-temporal resolution tradeoff over the entire image. Further, the technique in [11] requires long integration time for a single image capture, making it ill-suited for videos. The method presented in [9] simply rearranges/rebins the captured samples to produce different spatio-temporal resolutions, leading to visual artifacts due to aliasing. Our implementation allows choosing different resolutions for each image location independently, performs fast acquisition (results on dynamic scenes with up to 240 Hz), requires no masks, mitigates aliasing, and is simpler to implement with a regular camera, projector and a beam-splitter.

**Spatio-temporal super-resolution using multiple cameras:** Hybrid resolution imaging has been used for enhancing the resolution of videos with still

images [12], and for motion deblurring [13]. Wilburn et al. [14] used an array of cameras with temporally staggered short exposures to simulate a high-speed camera. Shechtman et al. [15] combined a set of videos captured at different spatial and temporal resolutions to achieve space-time super-resolution. Agrawal et al. [16] used multiple cameras with multiplexed coding for temporal super-resolution. All these techniques use multiple cameras for capturing videos at different resolutions that need to be decided pre-capture. The number of required cameras scales (at least linearly) with the required temporal speed-up. In contrast, our implementation requires only a single camera and projector, even for large temporal speed-ups.

## 2 Multi-resolution Sampling of the Space-Time Volume

In this section, we present our multi-resolution space-time sampling scheme. We show that this sampling can provide us with multiple spatio-temporal resolutions at each video location independently, using the same number of measurements as a conventional camera. Consider the group of 4 pixels in Figure 2a. We divide the integration time of each pixel into 4 equal intervals. Each of the 4 pixels is on for only one of the intervals (white indicates on, black indicates off). By switching on each pixel during a different time-interval, we ensure that each pixel samples the space-time volume at different locations.

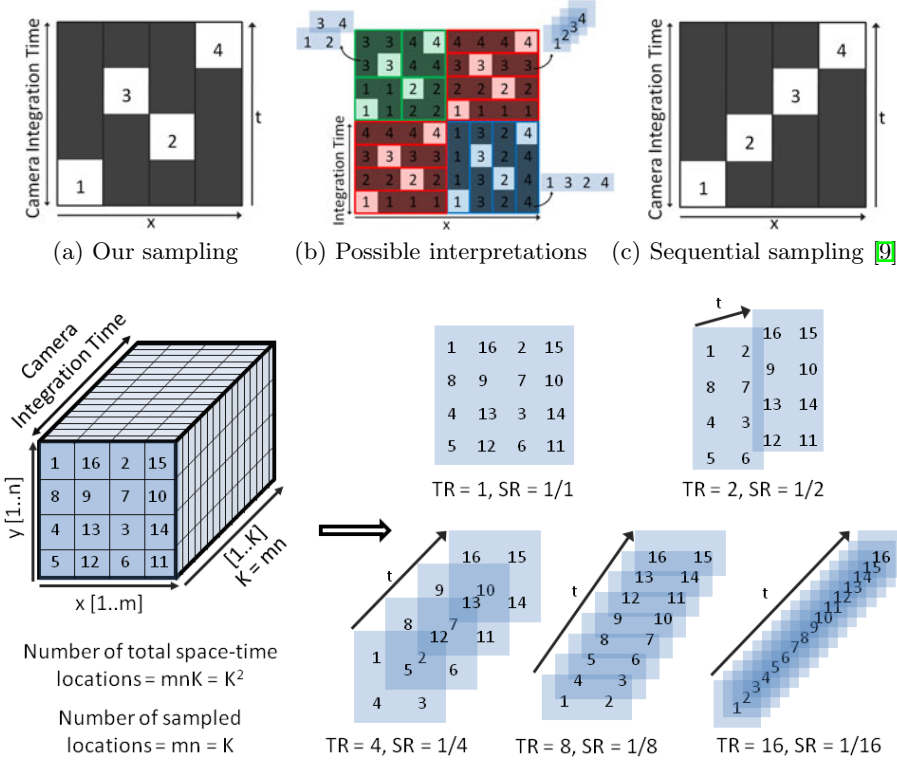
Different spatio-temporal resolutions can be achieved by simply re-binning these measurements, as illustrated in Figure 2b. For example, the four measurements can be arranged as temporal blocks (marked in red), spatial blocks (marked in blue) or as  $2 \times 2$  spatio-temporal blocks (marked in green). We define the [TR, SR] factors for a reconstruction as the gain in temporal and spatial resolution respectively over the acquired video. Thus, the [TR, SR] factors for these arrangements are  $[4, \frac{1}{4}]$ ,  $[1, \frac{1}{1}]$  and  $[2, \frac{1}{2}]$  respectively.

In general, consider the space-time volume  $V_{mn}$  defined by a neighborhood of  $m \times n$  pixels and one camera integration time, as illustrated in Figure 2, bottom-left. The integration time is divided into  $K = mn$  distinct sub-intervals, resulting in  $K^2$  distinct space-time locations. Different divisions of this volume into  $K$  equal rectilinear blocks correspond to different spatio-temporal resolutions. An illustration is shown in Figure 2. For the rest of the paper, we will use  $K$  for the pixel neighborhood size.

Each division of the volume corresponds to a spatio-temporal resolution<sup>1</sup>. A sampling scheme which facilitates all the resolutions corresponding to the different divisions should satisfy the following property: each block in every division must contain at least one measured sample. Since the total number of measured samples is only  $K$  (one for each pixel), each block will contain exactly one sample. Let  $x_p$  be the indicator variable for location  $p \in \{1, 2, \dots, K^2\}$ , such that  $x_p$  is 1 if the  $p^{\text{th}}$  location is sampled; it is 0 otherwise. Let  $B_{ij}$  be the  $i^{\text{th}}$  block in

---

<sup>1</sup> The total number of such divisions is the number of distinct factors of  $K$ . For example, for  $K = 16$ , we can have 5 distinct resolutions.

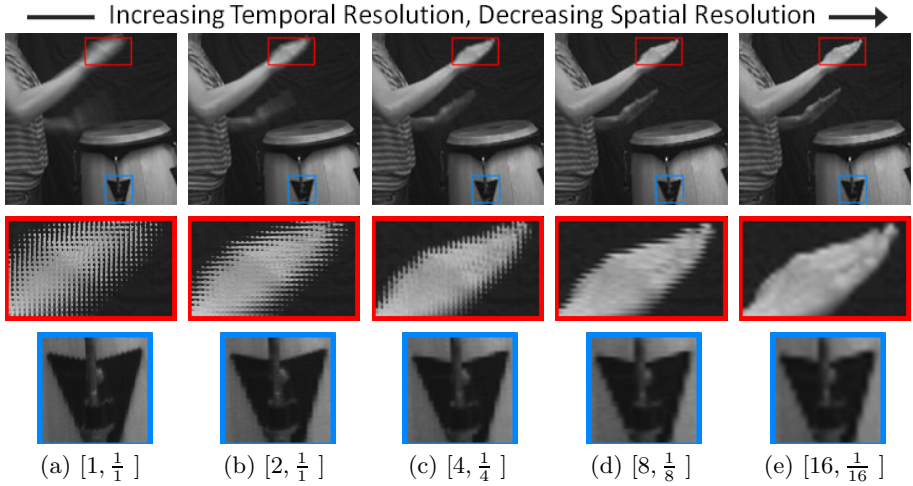


**Fig. 2. Simultaneously capturing multiple spatio-temporal resolutions:** (a) For a group of  $K$  neighboring pixels, each pixel is on for a temporal sub-segment of length  $\frac{1}{K}$  (white indicates on, black indicates off). For top row,  $K = 4$ . (b) These measurements can be interpreted post-capture as 4 temporal measurements (red), 4 spatial measurements (blue) or 4 spatio-temporal measurements (green). (c) Sequential sampling captures only a small sub-set of possible spatio-temporal resolutions. **Bottom row:** The temporal firing order for a group of  $4 \times 4$  pixels ( $K = 16$ ) and the possible resulting interpretations. With this sampling, we can achieve a temporal resolution gain of up to  $16X$ .

the  $j^{th}$  division of the volume. Then, for any pixel-neighborhood of a given size, a multi-resolution sampling can be computed by solving the following binary integer program:

$$\sum_{p \in B_{ij}} x_p = 1 \quad \forall B_{ij}, \quad \sum_{p=1}^{K^2} x_p = K, \quad x_p \in \{0, 1\} \quad \forall p \quad (1)$$

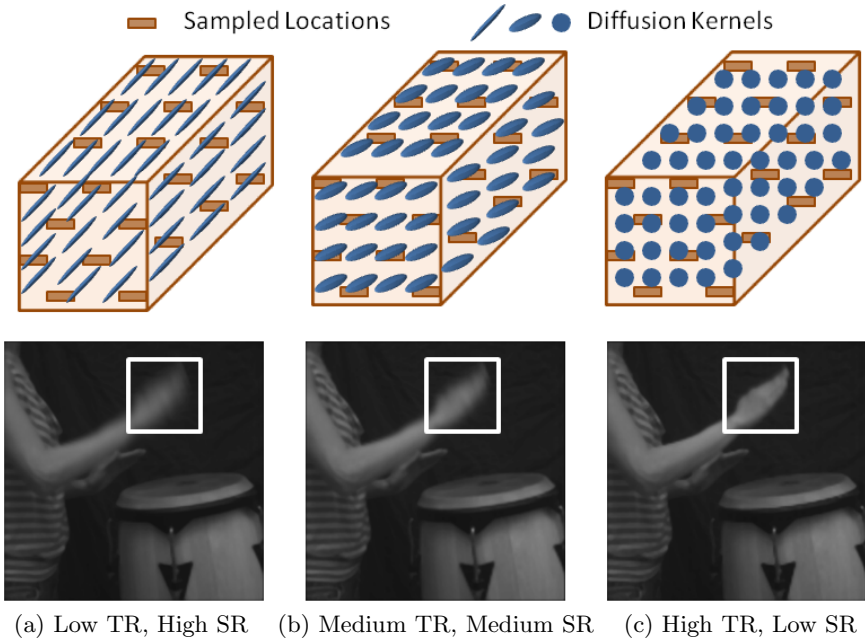
The first constraint ensures that every block in every division contains exactly one sample. The second constraint enforces the total number of samples to be equal to the number of pixels. For any given  $V_{mn}$ , the constraints can be generated automatically by computing different recti-linear divisions of the volume. The bottom row of Figure 2 shows the sampling order for a group of  $4 \times 4$  pixels



**Fig. 3. Generating multiple spatio-temporal resolutions by re-binning captured data:** (a) An image acquired with the temporal firing order given in Figure 2 bottom-left. The pixel neighborhood size is  $4 \times 4$ . (a-e) Different re-arrangements of the measurements, as given in Figure 2, and the corresponding [TR, SR] factors. From left to right, motion blur decreases but spatial resolution decreases as well. Simple re-binning of samples results in coded blur artifacts in the reconstructions.

computed by solving the integer program (II). The numbers denote the temporal firing order within an integration time. With this firing order, the samples can be arranged into 5 different spatio-temporal arrangements, shown on the bottom right. These arrangements correspond to resolutions with [TR, SR] factors of  $[1, \frac{1}{1}]$ ,  $[2, \frac{1}{2}]$ ,  $[4, \frac{1}{4}]$ ,  $[8, \frac{1}{8}]$  and  $[16, \frac{1}{16}]$  as compared to the acquired image. In contrast, sequential sampling [9] does not satisfy the constraints of the above binary integer program. As a result, it is amenable to a small sub-set of possible spatio-temporal resolutions. For the sequential sampling given in Figure 2c, the  $2 \times 2$  arrangement is not possible since not all the blocks are sampled.

**Simulating multi-resolution sampling:** To verify the feasibility of our multi-resolution sampling scheme, we used a Photron 1024 PCI camera to capture high-speed images at 480 Hz. The spatial resolution of the images is  $640 \times 480$ . The image is divided into neighborhoods of  $4 \times 4$  pixels. For each set of 16 consecutive frames, we weighted them according to the per-pixel code given on the bottom-left of Figure 2 and added them together. The resulting video is as if captured by a 30 Hz camera with a per-pixel shutter operating at 480 Hz. The scene consists of a person playing drums. While the hands move rapidly, the rest of the body moves slowly, and the drums move only on impact. An example image from the sequence is given in Figure 3 (top-left). Notice the per-pixel coded blur on the captured image (Figure 3a) as compared to usual smooth motion blur in regular cameras. This is because pixels encode temporal information as well. By rearranging the pixels according to Figure 2, we get sequences with different



**Fig. 4. Anisotropic diffusion for generating multiple spatio-temporal resolutions:** By interpolating the captured data with diffusion tensors of varying spectral shapes, we can achieve multiple spatio-temporal resolutions. The diffusion process also mitigates the effects of aliasing. Notice that coded blur artifacts are significantly reduced in comparison to the simple rebinning scheme of Figure 3.

combinations of spatio-temporal resolutions, as shown in Figure 3 (b-e) . From left to right, temporal resolution increases but the spatial resolution decreases.

### 3 Interpreting the Captured Data

In this section, we present post-capture algorithms for interpreting the data captured using our sampling scheme. One approach is simply re-arranging the measured samples to generate different spatio-temporal resolutions, as mentioned in the previous section. This scheme has two disadvantages: first, it restricts the possible spatio-temporal resolutions of the reconstructions to a few discrete choices. Second, it does not account for aliasing due to sub-sampling. Consequently, we witness disturbing visual artifacts such as coded blur (Figure 3) and temporal incoherence (pixel swimming). Such artifacts are specially noticeable in the presence of highly textured scene objects. In the following, we present a reconstruction algorithm which effectively addresses these limitations.

#### 3.1 Interpolation of Sub-sampled Data Using Anisotropic Diffusion

Let  $I_{(0)}$  be the initial space-time volume defined over a regular 3D grid. Our sampling scheme measures samples at a few locations in this volume. The re-

maining locations are considered missing data, as illustrated in Figure 4. We pose the reconstruction problem as *inpainting* the missing data by interpolating the measured samples using anisotropic diffusion [17][18]. The key idea is that by diffusing the intensities with tensors  $T$  of different spectral shapes (orientation), we can achieve different *effective* spatio-temporal resolutions. Consider the evolution of the image data with the number of iterations  $n$ :

$$\frac{\partial I}{\partial n} = \mathbf{trace}(TH) \text{ , where } H = \begin{bmatrix} I_{xx} & I_{xy} & I_{xt} \\ I_{yx} & I_{yy} & I_{yt} \\ I_{tx} & I_{ty} & I_{tt} \end{bmatrix} \quad (2)$$

is the  $3 \times 3$  Hessian matrix of the 3D image data  $I$ . The  $3 \times 3$  diffusion tensor defined by  $T = c_1\lambda\lambda^T + c_2\psi\psi^T + c_3\gamma\gamma^T$  [18] is characterized by its eigen values  $c_1, c_2, c_3$  and eigen vectors  $\lambda, \psi, \gamma$ . The solution of the PDE of Eqn. 2 is [18]:

$$I_{(n)} = I_{(0)} * G^{(T,n)} \text{ , where } G^{(T,n)}(\mathbf{x}) = \frac{1}{4\pi n} \exp\left(-\frac{\mathbf{x}^T T^{-1} \mathbf{x}}{4n}\right) \text{ ,} \quad (3)$$

where  $\mathbf{x} = (x \ y \ t)^T$ . Starting with the initial volume  $I_{(0)}$ , this PDE has the effect of progressively smoothing the data with oriented 3D Gaussians [2] defined by the tensor  $T$ . The PDE is repeatedly applied only on the missing data locations until the intensities from the measured samples diffuse to fill in the holes.

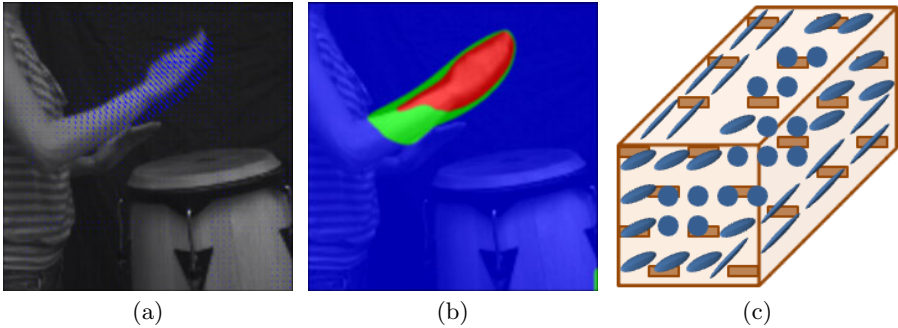
**A continuum of spatio-temporal resolutions:** By designing diffusion tensors of different spectral shapes, we can achieve different spatio-temporal resolutions of the reconstructed volume. Consider the set of axis-aligned ellipsoidal kernels  $T = \text{diag}(c_1, c_2, c_3)$ . If  $c_3 \gg c_1$  and  $c_3 \gg c_2$ , low-pass filtering occurs primarily in the temporal direction. Consequently, high-frequency content in the spatial direction is preserved. The resulting reconstruction, thus, has high spatial resolution and low temporal resolution, as illustrated in Figure 4a. On the other hand, if  $c_3 \ll c_1$  and  $c_3 \ll c_2$ , then most of the smoothing happens in the spatial direction, thus preserving high-frequency content in the temporal direction (Figure 4c). With  $c_1 = c_2 = c_3$ , the data is diffused isotropically in all three directions (Figure 4b). The reconstructions achieved with the simple scheme of re-arranging samples correspond to special cases of the diffusion tensor. For example, the  $[1, \frac{1}{1}]$  reconstruction can be achieved by using a tensor with  $c_1 = c_2 = 0, c_3 = 1$ . Similarly, with  $c_1 = c_2 = 1, c_3 = 0$ , we can achieve the  $[16, \frac{1}{16}]$  reconstruction.

**Aliasing artifacts:** The diffusion process interpolates and regularizes the data on the 3D grid, thus mitigating the effects of aliasing due to sub-sampling. Consequently, coded blur and temporal coherence artifacts are significantly reduced in the reconstructions. **See the project web-page [19] for comparisons.**

---

<sup>2</sup> An equivalent representation of the tensor  $T$  is in terms of oriented ellipsoids.





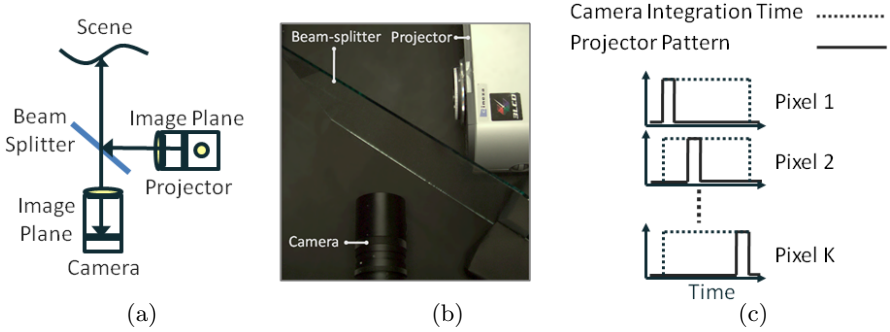
**Fig. 5. Motion-aware video reconstruction:** (a) Quiver plot of the optical flow between two successive frames of a high TR reconstruction. (b) Color coded magnitude of the optical flow. Red indicates fast moving objects, green indicates slow moving and blue indicates stationary objects. Raw data is interpolated with diffusion tensors oriented along the optical flow vectors (c) to achieve a motion aware reconstruction. The resulting frame is shown in Figure III.

## 4 Motion-Aware Video

The reconstruction algorithms discussed so far are independent of the captured data, which, although sparse, can provide useful information about the scene. In this section, we present an algorithm to use the motion information in the captured data to drive the reconstruction process. We call the resulting reconstruction *motion-aware*: the spatio-temporal resolution trade-off at each location is resolved according to the motion information at that location. Such a reconstruction would minimize the motion blur for fast moving objects while simultaneously maximizing the spatial frequency content for slow moving or static objects. Following is the algorithm we use for computing such a reconstruction:

**Step 1: High TR reconstruction:** It can be extremely difficult to recover faithful motion information in the presence of large motion blur. Thus, a high temporal resolution reconstruction is imperative for computing accurate motion information. Our first step is to do a high TR reconstruction using an axis-aligned tensor  $T = \text{diag}(c_1, c_2, c_3)$  with  $(c_1, c_2, c_3) = (1.0, 1.0, 0.05)$ . Such a reconstruction would smooth primarily in the spatial dimensions, thus preserving high-frequency temporal content. A small value is assigned to  $c_3$  to mitigate temporal flickering artifacts.

**Step 2: Computing optical flow:** We compute motion information in the form of optical flow between successive frames of the high TR reconstruction. For this, we used an implementation of the optical flow method given by Brox et al [20]. Since computed on a high TR reconstruction, the optical flow estimates are fairly robust, even for fast moving objects. Figures 5a and 5b illustrate the optical flow between two successive frames of the drums sequence using a quiver plot and color coded magnitudes respectively. Red indicates fast moving objects, green indicates slow moving and blue indicates stationary objects. Although the



**Fig. 6. Hardware setup for simulating per-pixel shutter:** (a-b) Our setup consists of co-locating and temporally synchronizing a camera (15 Hz) and a projector (240 Hz). Under no global illumination, a camera pixel receives light only when the corresponding projector pixel is on. (c) The observed irradiance at a camera pixel is modulated according to the binary pattern on the corresponding projector pixel.

optical flow vectors have high temporal resolution, their spatial resolution is much lesser than that of the scene itself. Thus, computing optical flow at a low spatial resolution does not result in significant spatial aliasing. In contrast, optical flow estimates on the original captured data are unreliable due to the presence of large, albeit coded motion blur.

**Step 3: Motion driven diffusion:** The key idea is to design diffusion tensors at each location so that they smooth along the motion direction. Let  $(u, v, 1)$  be the optical flow vector at a given location. We define the diffusion tensor as  $T = c_1 \lambda \lambda^T + c_2 \psi \psi^T + c_3 \gamma \gamma^T$ , where

$$\lambda = \frac{(u, v, 1)}{\sqrt{u^2 + v^2 + 1}}, \quad \psi = \lambda \times (0, 0, 1), \quad \gamma = \lambda \times \psi \quad (4)$$

form an ortho-normal set of unit vectors. By choosing  $c_1 = 0.95, c_2 = 0.05, c_3 = 0.05$ , we orient the diffusion tensor sharply along  $\lambda$ , the motion direction. Note that this results in a **variable** diffusion tensor field over the space-time volume (Figure 5c) as different locations have different optical flow vectors. An example frame from the motion-aware reconstruction of the drums sequence is given in Figure 1f. Note that the motion blur is minimized on the fast moving hands while the drums and the body retain high spatial resolution. Results with real experimental data are given in Figures 7 and 8.

## 5 Hardware Implementation of Per-Pixel Shutter

The sampling scheme discussed in the previous sections requires a fast ( $K$  times the frame-rate of the camera) per-pixel shutter on the sensor array. Currently

available cameras have fast global shutters<sup>3</sup> implemented as external trigger modes [22]. However, these modes do not provide per-pixel control. Recently, DMD arrays have been used to provide precise, per-pixel temporal modulation [9,23]. These devices are commonly used as light modulators in off-the shelf DLP projectors. We have implemented per-pixel shutter using a DLP projector in conjunction with a camera. The projector is used to provide fast, per-pixel light modulation externally.

The projector and the camera are *co-located* using a beam-splitter, as shown in Figure 6. The setup is placed in a dark room. We assume that there is no ambient or global illumination. Co-location is achieved by aligning the camera and the projector so that the camera does not observe any shadows cast by the projector. This procedure takes about 15 mins. Co-location ensures that the camera and the projector image planes are related by a single homography irrespective of the scene.

The camera and the projector are temporally synchronized so that for each camera integration time, the projector cycles through  $K$  binary patterns. The binary patterns consist of tiles of  $K$  pixels repeated spatially. Each tile encodes the sampling scheme being used. Since there is no ambient illumination, a camera pixel receives light only when the corresponding projector pixel is on. Consequently, the irradiance at a camera pixel is modulated according to the binary pattern on the corresponding projector pixel. An illustration is shown in Figure 6c. This modulation acts as per-pixel shutter. The temporal frequency of modulation (hence the shutter), is given by the frame rate of the projector.

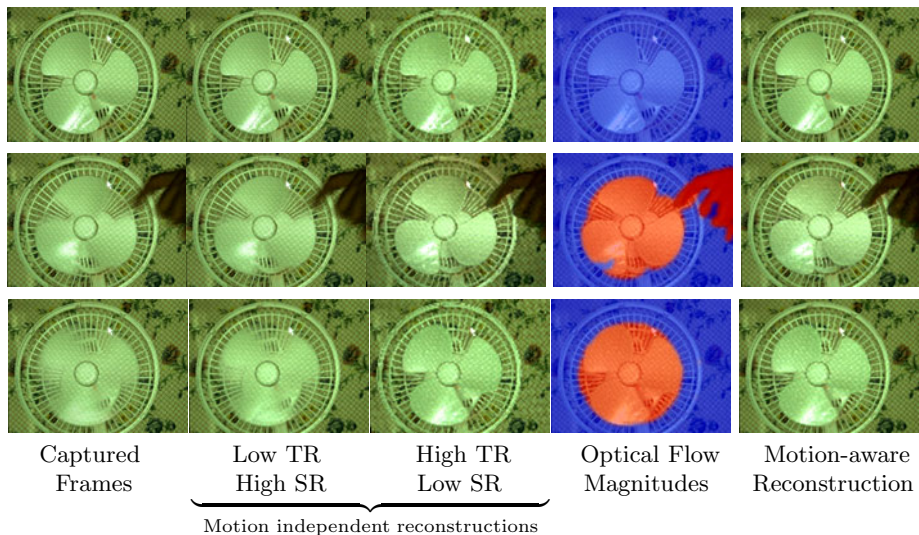
We used a Point-Grey Flea2 camera and a Multi-Use-Light-Engine (MULE) projector [24]. With a 60 Hz video input, the MULE projector can project binary bit-planes at up to  $60 \times 24 = 1440$  Hz. To implement the coding scheme given in Figure 3a, we operated the projector at 240 Hz, thus achieving a frame-rate of 240 Hz even though the frame rate of the camera is 15 Hz.

## 5.1 Real Experiments and Results

**Fan rotating scene (Figure 7):** The first sequence consists of a rotating fan acquired with a camera running at 7.5 Hz. The frames have significant motion blur and temporal aliasing. In this case, the pixel neighborhood size was  $2 \times 4$ ; thus,  $K = 8$ . The second and the third columns show 1 frame each from two reconstructions done with the diffusion tensors  $T = \text{diag}(0.05, 0.05, 1)$  and  $T = \text{diag}(1, 1, 0.05)$  respectively. We call these motion-independent reconstructions, as these reconstructions do not use any motion information. The high TR reconstruction has a temporal resolution of  $7.5 \times 8 = 60$  Hz. The fourth column shows optical flow magnitudes between two successive frames of the high TR reconstruction. The optical flow information is used for computing a motion-aware reconstruction (last column), as discussed in Section 4.

---

<sup>3</sup> Fast global shutters have been used in the past for motion deblurring [21], but modulate all pixels simultaneously.



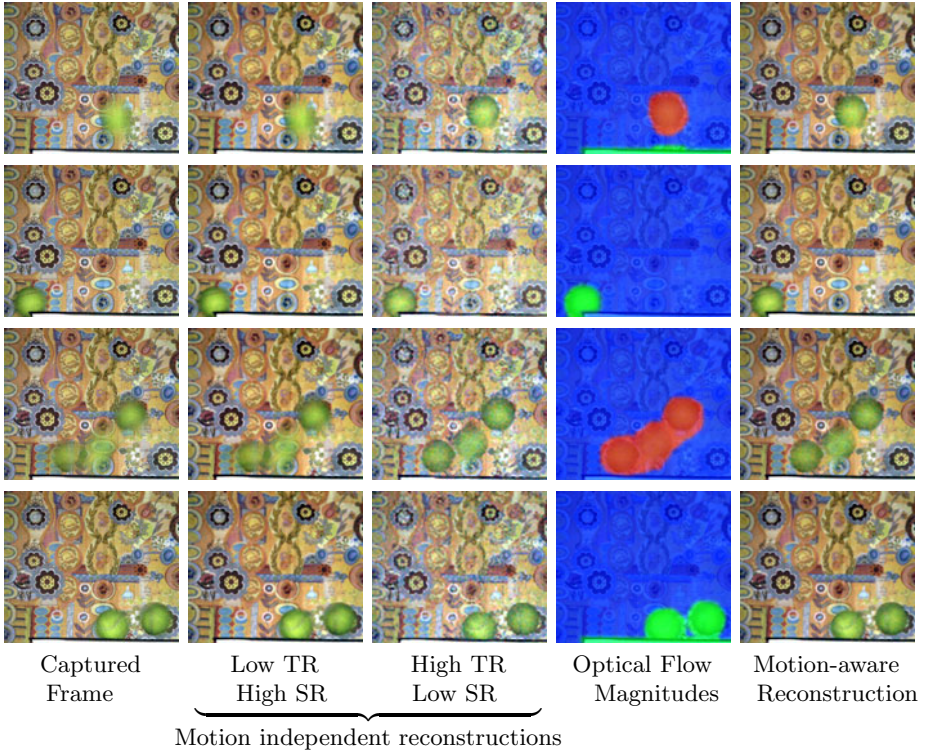
**Fig. 7. Motion-aware video of rotating fan:** (First column) Raw frames from the captured sequence. (Second and the third columns) One frame each from two reconstructions done with different diffusion tensors. (Fourth column) Optical flow magnitudes between two successive frames of the high TR reconstruction. (Last column) Motion aware reconstruction. Notice the much reduced motion blur on the fan and high-spatial resolution on the static background. Zoom in for details.

**Multiple Balls Bouncing (Figure 8):** This sequence consists of multiple balls colliding with each other at high velocities. The camera is running at 15 Hz. We used a pixel neighborhood of  $4 \times 4$ ; thus,  $K = 16$ . The second and third columns show one frame each from reconstructions with tensors  $T = \text{diag}(0.05, 0.05, 1)$  and  $T = \text{diag}(1, 1, 0.05)$  respectively. The last column shows motion-aware reconstruction. Notice that one of the balls is almost invisible in the captured frame of third row due to large motion blur. In the motion aware reconstruction, it can be easily localized.

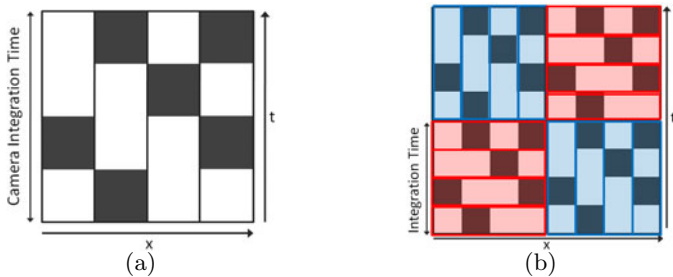
## 6 Discussion and Limitations

The goal of this work was to build video cameras whose spatial and temporal resolutions can be changed post-capture depending on the scene. We have presented the first example of an imaging system which allows multiple space-time resolutions at each image location independently - using programmable, fast per-pixel shutters and a content-aware post-processing scheme.

A limitation of our sampling scheme is that the pixels collect light over only a fraction of the integration time leading to low signal-to-noise ratio (SNR). The trade-off between temporal resolution and SNR is well known for video cameras. High-speed cameras suffer from significant image noise in low-light



**Fig. 8. Motion-aware video of multiple bouncing balls:** (First column) Raw frames from the captured sequence. (Second-third columns) One frame each from two reconstructions done with different diffusion tensors. (Fourth column) Optical flow magnitudes between two successive frames of the highest TR reconstruction. (Last column) Motion aware reconstruction.



**Fig. 9. Multiplexed sampling:** By using multiplexed codes (a), each pixel gathers more light resulting in higher SNR (white indicates on, black indicates off). Post-capture reshaping of voxels (b) can be achieved by de-multiplexing the captured data.

conditions. This trade-off can be countered by incorporating multiplexing into our sampling scheme. With multiplexed codes, as shown in Figure 9a, each pixel gathers more light as compared to identity codes (Figure 2a). This is similar in spirit to capturing images using multiplexed illumination for achieving higher SNR [25]. Post-capture reshaping of voxels can be achieved by de-multiplexing.

Our implementation of per-pixel shutter using a projector-camera system is limited to scenes with low global and ambient illumination. Passive implementations using either a DMD array [9,23] or variable integration on sensor chip can effectively address these limitations.

**Acknowledgments.** This research was supported in parts by ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762, Okawa Foundation Grant and NSF CAREER award IIS-0643628 and Mitsubishi Electrical Research Labs. Authors thank Jinwei Gu and Shree K. Nayar for use of the MULE projector.

## References

1. Zabrodsky, H., Peleg, S.: Attentive transmission. *J. of Visual Comm. and Image Representation* 1 (1990)
2. Baraniuk, R.: Compressive sensing. *IEEE Signal Processing Magazine* 24 (2007)
3. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. *ACM Trans. Graph* 28 (2009)
4. Nayar, S.K., Mitsunaga, T.: High dynamic range imaging: spatially varying pixel exposures. In: *IEEE CVPR* (2000)
5. Narasimhan, S.G., Nayar, S.K.: Enhancing resolution along multiple imaging dimensions using assorted pixels. *PAMI* 27 (2005)
6. Ng, R.: Fourier slice photography. *ACM Trans. Graphics* 24, 735–744 (2005)
7. Horstmeyer, R., Euliss, G., Athale, R., Levoy, M.: Flexible multimodal camera using a light field architecture. In: *ICCP* (2009)
8. Ben-Ezra, M., Zomet, A., Nayar, S.K.: Video super-resolution using controlled subpixel detector shifts. *PAMI* 27 (2005)
9. Bub, G., Tecza, M., Helmes, M., Lee, P., Kohl, P.: Temporal pixel multiplexing for simultaneous high-speed, high-resolution imaging. *Nature Methods* (2010)
10. Gu, J., Hitomi, Y., Mitsunaga, T., Nayar, S.K.: Coded rolling shutter photography: Flexible space-time sampling. In: *ICCP* (2010)
11. Agrawal, A., Veeraraghavan, A., Raskar, R.: Reinterpretable imager: Towards variable post capture space, angle & time resolution in photography. In: *Eurographics* (2010)
12. Gupta, A., Bhat, P., Dontcheva, M., Curless, B., Deussen, O., Cohen, M.: Enhancing and experiencing spacetime resolution with videos and stills. In: *ICCP* (2009)
13. Ben-Ezra, M., Nayar, S.K.: Motion-based motion deblurring. *PAMI* 26 (2004)
14. Wilburn, B., Joshi, N., Vaish, V., Levoy, M., Horowitz, M.: High speed video using a dense camera array. In: *IEEE CVPR* (2004)
15. Shechtman, E., Caspi, Y., Irani, M.: Space-time super-resolution. *PAMI* 27 (2005)
16. Agrawal, A., Gupta, M., Veeraraghavan, A., Narasimhan, S.G.: Optimal coded sampling for temporal super-resolution. In: *IEEE CVPR* (2010)

17. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE PAMI* 12 (1990)
18. Tschumperle, D., Deriche, R.: Vector-valued image regularization with pdes: A common framework for different applications. *PAMI* 27 (2005)
19. Project web-page, <http://graphics.cs.cmu.edu/projects/FlexibleVoxels/>
20. Brox, T., Bruhn, A., Papenber, 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)
21. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graphics* 25, 795–804 (2006)
22. External trigger modes supported by point grey cameras, <http://www.ptgrey.com/support/kb/>
23. Nayar, S.K., Branzoi, V., Boulton, T.E.: Programmable imaging: Towards a flexible camera. In: *IJCV* (2006)
24. McDowall, I., Bolas, M.: Fast light for display, sensing and control applications. In: *IEEE VR 2005 Workshop on Emerging Display Technologies* (March 2005)
25. Schechner, Y., Nayar, S.K., Belhumeur, P.: A theory of multiplexed illumination. In: *ICCV* (2003)

# Learning PDEs for Image Restoration via Optimal Control

Risheng Liu<sup>1</sup>, Zhouchen Lin<sup>2,\*</sup>, Wei Zhang<sup>3</sup>, and Zhixun Su<sup>1</sup>

<sup>1</sup> Dalian University of Technology, Dalian 116024, P.R. China  
rsliu@mail.dlut.edu.cn, zxsu@dlut.edu.cn

<sup>2</sup> Microsoft Research Asia, Beijing 100190, P.R. China  
zhoulin@microsoft.com

<sup>3</sup> The Chinese University of Hong Kong, Shatin, Hong Kong, P.R. China  
zw009@ie.cuhk.edu.hk

**Abstract.** Partial differential equations (PDEs) have been successfully applied to many computer vision and image processing problems. However, designing PDEs requires high mathematical skills and good insight into the problems. In this paper, we show that the design of PDEs could be made easier by borrowing the *learning strategy* from machine learning. In our learning-based PDE (L-PDE) framework for image restoration, there are two terms in our PDE model: (i) a regularizer which encodes the prior knowledge of the image model and (ii) a linear combination of differential invariants, which is data-driven and can effectively adapt to different problems and complex conditions. The L-PDE is learnt from some input/output pairs of training samples via an optimal control technique. The effectiveness of our L-PDE framework for image restoration is demonstrated with two exemplary applications: image denoising and inpainting, where the PDEs are obtained easily and the produced results are comparable to or better than those of traditional PDEs, which were elaborately designed.

## 1 Introduction

### 1.1 Prior Work

Partial differential equations (PDEs) have been successfully applied to solve many problems in computer vision and image processing. This kind of methods can date back to the 1960s [1][2]. However, this technique did not draw much attention until the introduction of the concept of scale space by Koenderink [3] and Witkin [4] in the 1980s. The Perona-Malik (P-M) anisotropic equation [5] and the mean curvature motion (MCM) equation [6] further drew great interest from researchers toward designing PDEs for various problems in computer vision and image processing. In general, there are two types of methods for designing PDEs for vision tasks [7]:

1. **Variational Design:** Basically, variational methods first define an energy functional to collect the desired properties of the output image, including the image prior models (e.g., the Tikhonov regularizer [8] and the total variation (TV) regularizer [9]), and then derive the evolution equations by computing the Euler-Lagrange equation of the energy functional.

---

\* Corresponding author.



2. **Direct Design:** Direct methods involve writing down the PDEs directly, based on the mathematical and physical understandings of the problem. This method requires proficiency in the properties of differential operators, in particular nonlinear ones. Famous examples include anisotropic diffusion [5], shock filter [10] and curve evolution [6].

In a geometric view, most traditional PDEs in computer vision and image processing are obtained by either optimizing some global geometric quantities (e.g., length, area, and total squared curvature) or by computing geometric invariants under certain transformation groups. All of these methods require good skills when choosing appropriate PDE forms and predicting the final effect of composing related terms such that the obtained PDEs roughly meet the goals. A lot of trial and error may also be necessary for designing a good PDE. As a result, current methods for designing PDEs greatly limit the applications of PDEs to wider and more complex scopes. This motivates us to explore whether we can acquire PDEs that are more powerful but require much less human effort.

## 1.2 Our Approach

Inspired by learning-based methods in machine learning, we would like to explore a framework for learning PDEs to accomplish various computer vision and image processing tasks. In this paper, as preliminary work, we propose a learning-based PDE (L-PDE) framework for image restoration. For image restoration problems, we know that the output image should obey some statistical models of natural images. Such statistical models can serve as the regularizer term in our PDE, which controls the output image, making it a natural image. Hence this term is called the *regularization* term. The other term in our PDE is to cope with different image restoration problems and different data. As most image restoration problems are translationally and rotationally invariant, i.e., when the input image is translated or rotated by some amount the output image is also translated or rotated by the same amount, this second term must be functions of fundamental differential invariants [11] that are invariant under translation and rotation. We assume that the second term is a linear combination of the fundamental differential invariants. Although a linear combination is simple, our PDE model is already general enough and many existing PDEs can be viewed as a special case of our model. The linear combination coefficients are learnt from real data so that the learnt PDE can adapt to different image restoration problems and different data. Hence the second term is called the *data-driven differential invariant* term.

To learn the coupling coefficients among the differential invariants in the data-driven term, we prepare some input/output training image pairs and adopt a technique called PDE-based optimal control [12]. Once the coefficients are computed, the L-PDE is obtained and can be applied to test images. Hence with our framework, the most effort on obtaining a PDE is preparing some input/output training image pairs. So our L-PDE framework might be a possible way of designing PDEs for vision tasks in a lazy manner. Though the optimal control technique has already been applied to some computer vision problems, such as optical flow estimation [13] and tracking [14], we use it in a different way. We aim at determining the form (coefficients) of the PDEs, while the existing

work uses the optimal control to determine the outputs of their PDEs, which are known *a priori*. In short, *our L-PDE framework connects PDE-based methods and learning-based methods via optimal control.*

## 2 Learning-Based PDE Model

In this section, we present the form of the PDEs in our L-PDE framework for image restoration. We denote  $f$  as the input image and  $u$  as the desired output image. The meaning of the notations that will be used hereafter can be found in Table 1.

**Table 1.** Notations

$\Omega$	An open bounded region of $\mathbb{R}^2$	$\partial\Omega$	Boundary of $\Omega$
$(x, y)$	$(x, y) \in \Omega$ , spatial variable	$t$	$t \in (0, T_f)$ , temporal variable
$Q$	$\Omega \times (0, T_f)$	$\Gamma$	$\partial\Omega \times (0, T_f)$
$\ \cdot\ $	$L^2$ norm	$\nabla u$	Gradient of $u$
$\mathbf{H}_u$	Hessian of $u$	$\text{div}(\mathbf{u})$	Divergence of $\mathbf{u}$
$\wp$	$\wp = \{(0, 0), (0, 1), (1, 0), (0, 2), (1, 1), (2, 0)\}$ , index set for differentiation		
$\kappa(u)$	$\kappa(u) = \text{div}\left(\frac{\nabla u}{\ \nabla u\ }\right)$ , mean curvature of $u$		

### 2.1 Description of Our PDE Model

Our PDE model is an evolutionary PDE combining a TV regularizer and a linear combination of fundamental differential invariants:

$$\begin{cases} \frac{\partial u}{\partial t} = L(u, \mathbf{a}), & (x, y, t) \in Q, \\ u = 0, & (x, y, t) \in \Gamma, \\ u|_{t=0} = f, & (x, y) \in \Omega, \end{cases} \quad (1)$$

where  $L(u, \mathbf{a}) = \kappa(u) + F(u, \mathbf{a})$ . The Dirichlet boundary condition<sup>1</sup> is for ease of mathematical deduction. The forms and the geometric meanings of  $\kappa(u)$  and  $F(u, \mathbf{a})$  will be presented below.

**Total Variation Regularization Term:** The TV regularization has been successfully incorporated in PDEs for a large class of computer vision and image processing problems due to its mathematical tractability and effectiveness in representing the statistical model of natural images. It was first introduced to computer vision and image processing by Rudin, Osher and Fatemi (ROF) in their paper on edge preserving image denoising [9]. It first defines a variational minimization model  $\min_u \int_{\Omega} \|\nabla u\| d\Omega$  in the bounded variation space (which allows for piecewise constant images) and then derives the mean curvature  $\kappa(u)$  as the regularization term in its associated Euler-Lagrange equation [9, 7]. The TV regularization is especially useful in applications, e.g., image restoration, where edges are to be respected. That is why our PDE model incorporates  $\kappa(u)$ .

<sup>1</sup> As in real applications  $f$  will be padded with zeros of several pixels' width around the input image, the difference between the Dirichlet boundary condition in our model and the Neumann boundary condition in traditional PDEs is slight.

**Table 2.** The fundamental differential invariants up to the second order

$\mathbf{inv}(u) = [\mathbf{inv}_0(u), \dots, \mathbf{inv}_5(u)]^T$		
$i$	$\mathbf{inv}_i(u)$	
0	$f$	
1	$u$	
2	$\ \nabla u\ ^2 = u_x^2 + u_y^2$	Zeroth Order
3	$\text{tr}(\mathbf{H}_u) = u_{xx} + u_{yy}$	Second Order
4	$\text{tr}(\mathbf{H}_u^2) = u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2$	
5	$(\nabla u)^T \mathbf{H}_u (\nabla u) = u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}$	

**Data-Driven Differential Invariant Term:** As we have explained in Section 1.2, the data-driven differential invariant term is a linear combination of fundamental differential invariants that are invariant under translation and rotation. For 2D scalar images, there are five such fundamental differential invariants up to the second order [11]. They are listed in Table 2 ( $f$  is added as the zeroth invariant due to the following geometric considerations). All the differential invariants have geometric meanings.  $\mathbf{inv}_0(u) = f$  is the input image.  $\mathbf{inv}_1(u) = u$  is the desired output image.  $\mathbf{inv}_2(u) = \|\nabla u\|^2$  is the squared norm of the gradient.  $\mathbf{inv}_3(u) = \text{tr}(\mathbf{H}_u)$  is the Laplacian, which has been widely used to measure the smoothness of an image [15].  $\mathbf{inv}_4(u) = \text{tr}(\mathbf{H}_u^2)$ , known as “deviation from flatness”, is another useful way to measure the local “unflatness” of the image.  $\mathbf{inv}_5(u) = (\nabla u)^T \mathbf{H}_u (\nabla u)$  is a kind of image “curvature”, which has been used as a general purpose visual front-end operation [16]. Using such differential invariants, all local intrinsic properties of images, which should be invariant to coordinate transformation, can be described. Therefore, the data-driven term for our L-PDE model can be written as:

$$F(u, \mathbf{a}) = \mathbf{a}(t)^T \mathbf{inv}(u), \quad (2)$$

where  $\mathbf{a}(t) = [a_0(t), \dots, a_5(t)]^T$  are coefficient functions, which are used to control the evolution of  $u$ . For different problems,  $\mathbf{a}(t)$  is different. They are learnt from training images and hence our L-PDE can adapt to different problems and data. We will present a PDE-based optimal control framework to learn these coefficient functions in Section 3.

## 2.2 Connection between L-PDE and Traditional PDEs

In this subsection, we discuss the relationship between our L-PDE model and some well-known related work.

Traditional PDEs were designed with different insights. However, as shown in Table 3, many of those for image restoration in fact share a common formulation and are all special cases of our proposed L-PDE model. The difference between these PDEs lies in the choice of  $\mathbf{a}(t)$  and the regularization term. However, our L-PDE model and the traditional PDEs present intrinsically different perspectives on interpreting the form of PDEs. Traditional PDEs are all crafted by people with skills, based on their insight to the problems, whereas our model automatically determines the PDEs from real data. One can easily see that manually designed PDEs only correspond to trivial coefficient functions, where only popular differential invariants, e.g., the Laplacian and the zeroth

**Table 3.** Reformulating some popular PDEs in our L-PDE model

PDE	$\mathbf{a}(t)$ in data-driven term	Regularization term
Gaussian scale space [3]	$\mathbf{a}(t) = [0, 0, 0, 1, 0, 0]^T$	–
Tikhonov [8]	$\mathbf{a}(t) = [1, -1, 0, 1, 0, 0]^T$	–
ROF [9], TV inpainting [17]	$\mathbf{a}(t) = [1, -1, 0, 0, 0, 0]^T$	$\kappa(u)$

order invariants, are used. Moreover, the nonzero coefficients are also *special constants*. In comparison, the coefficients in our L-PDEs can be much more flexible. They may not be sparse. They can be arbitrary real numbers and can even vary with time. So our L-PDE model can be much more adaptive to the input images and solve different image restoration problems in a unified framework.

### 3 Learning Coefficients via Optimal Control

#### 3.1 The Objective Functional

Given the form of the general data-driven term in (2), we have to determine the coefficient functions  $\mathbf{a}(t)$  in order to obtain a workable PDE. We may prepare some pairs of input/output training samples  $(f_k, \tilde{u}_k)$ , where  $f_k$  is the input image and  $\tilde{u}_k$  is the expected output image. Since the final output of our PDE should be close to the ground truth, the coefficient functions should minimize the following functional:

$$J(\{u_k\}_{k=1}^K, \mathbf{a}) = \frac{1}{2} \sum_{k=1}^K \int_{\Omega} (u_k(T_f) - \tilde{u}_k)^2 d\Omega + \frac{1}{2} \sum_{i=0}^5 \alpha_i \int_0^{T_f} a_i^2(t) dt, \quad (3)$$

where  $u_k(T_f)$  is the output image at time  $t = T_f$  computed from (1) when the input image is  $f_k$ , and  $\alpha_i$  are positive weighting parameters<sup>3</sup>. The first term of  $J$  requires the final output of our PDE to be close to the ground truth. The second term is for regularization so that this optimal control problem is well-posed.

#### 3.2 Solving the Optimal Control Problem

Then we have the following optimal control problem with PDE constraints:

$$\min_{\mathbf{a}} J(\{u_k\}_{k=1}^K, \mathbf{a}), \quad s.t. \begin{cases} \frac{\partial u_k}{\partial t} = L(u_k, \mathbf{a}), & (x, y, t) \in Q, \\ u_k = 0, & (x, y, t) \in \Gamma, \\ u_k|_{t=0} = f_k, & (x, y) \in \Omega. \end{cases} \quad (4)$$

By introducing the adjoint equation of (4), the Gâteaux derivative of  $J$  can be computed and consequently, the (locally) optimal  $\mathbf{a}(t)$  can be computed via gradient based

<sup>2</sup> For different problems,  $T_f$  may be different. How to determine the optimal  $T_f$  is left to future work.

<sup>3</sup> In this paper, we simply fix  $\alpha_i = 10^{-7}$ ,  $i = 0, \dots, 5$ .

algorithms (e.g., conjugate gradient [18]). Here, we give the adjoint equation and Gâteaux derivative directly due to the page limit<sup>4</sup>.

**Adjoint Equation:** The adjoint equation of (4) is:

$$\begin{cases} \frac{\partial \varphi_k}{\partial t} + \sum_{(p,q) \in \wp} (-1)^{(p+q)} \frac{\partial^{p+q} (\sigma_{pq}(u_k) \varphi_k)}{\partial x^p \partial y^q} = 0, & (x, y, t) \in Q, \\ \varphi_k = 0, & (x, y, t) \in \Gamma, \\ \varphi_k|_{t=T_f} = \tilde{u}_k - u_k(T_f), & (x, y) \in \Omega, \end{cases} \quad (5)$$

where

$$\sigma_{pq}(u) = \frac{\partial L(u)}{\partial u_{pq}} = \frac{\partial \kappa(u)}{\partial u_{pq}} + \sum_{i=0}^5 a_i \frac{\partial \text{inv}_i(u)}{\partial u_{pq}} \quad \text{and} \quad u_{pq} = \frac{\partial^{p+q} u}{\partial x^p \partial y^q}.$$

**Gâteaux Derivative of the Functional:** With the help of the adjoint equation, at each iteration the derivative of  $J$  with respect to  $\mathbf{a}(t)$  is as follows:

$$\frac{\partial J}{\partial a_i} = \alpha_i a_i - \sum_{k=1}^K \int_{\Omega} \varphi_k \text{inv}_i(u_k) d\Omega, \quad i = 0, \dots, 5. \quad (6)$$

where the adjoint function  $\varphi_k$  is the solution to (5).

### 3.3 Initialization of $\mathbf{a}(t)$

A good initialization of  $\mathbf{a}(t)$  results in a better approximation power of the learnt PDE and also makes the optimization process shorter. Here we propose a heuristic method for initializing the coefficient functions. At each time step,  $\frac{\partial u_k(t)}{\partial t}$  is expected to be  $\frac{\tilde{u}_k - u_k(t)}{T_f - t}$  so that  $u_k$  tends to the expected output  $\tilde{u}_k$ . On the other hand, with  $\frac{\partial u_k(t)}{\partial t} = L(u_k, \mathbf{a})$ , we want  $\mathbf{a}(t)$  to minimize:

$$\sum_{k=1}^K \int_{\Omega} \left( L(u_k, \mathbf{a}) - \frac{\partial u_k(t)}{\partial t} \right)^2 d\Omega = \sum_{k=1}^K \int_{\Omega} [\mathbf{p}_k(t)^T \mathbf{a}(t) - d_k(t)]^2 d\Omega, \quad (7)$$

where  $\mathbf{p}_k(t) = \text{inv}(u_k)$  and  $d_k(t) = \frac{\tilde{u}_k - u_k(t)}{T_f - t} - \kappa(u_k)$ . So the initial  $\mathbf{a}(t)$  can be obtained by solving the following system<sup>5</sup>:

$$\mathbf{P}(t) \mathbf{a}(t) = \mathbf{d}(t), \quad (8)$$

where  $\mathbf{P}(t) = \sum_{k=1}^K \int_{\Omega} \mathbf{p}_k(t) \mathbf{p}_k(t)^T d\Omega$  and  $\mathbf{d}(t) = \sum_{k=1}^K \int_{\Omega} \mathbf{p}_k(t) d_k(t) d\Omega$ .

<sup>4</sup> For more details and a more mathematically rigorous exposition, please see Supplementary Material and refer to [19,20,21].

<sup>5</sup> For notational convenience, we simply write integrals here. In real computation, the integrals should be discretized.

## 4 Our L-PDE Framework for Image Restoration

We now summarize our L-PDE framework for image restoration in Algorithm 1. After the PDE is learnt, it can be applied to new test images by solving (1), whose input  $f$  is the test image and the solution  $u(T_f)$  is the desired output image.

---

### Algorithm 1. (The framework to learn PDEs for image restoration)

---

**Require:** Training image pairs  $(f_k, \tilde{u}_k)$ ,  $k = 1, \dots, K$ ;  $T_f$ .

- 1: Initialize  $\mathbf{a}(t)$ ,  $t \in [0, T_f]$ , by solving (8).
- 2: **while** not converged **do**
- 3:   Compute  $\frac{\partial J}{\partial a_i}$ ,  $i = 0, \dots, 5$ , using (6).
- 4:   Decide the search direction using the conjugate gradient method [18];
- 5:   Perform golden search along the search direction and update  $\mathbf{a}(t)$ .
- 6: **end while**

**Ensure:** The coefficient functions  $\mathbf{a}(t)$ ,  $t \in [0, T_f]$ .

---

## 5 Experimental Results

In this section, we demonstrate the applications of our L-PDE framework for image restoration to two problems, denoising and inpainting. Our experiments are done on grayscale images. *For the best visual comparison, the readers are encouraged to inspect the images in this section on screen.*

### 5.1 Image Denoising

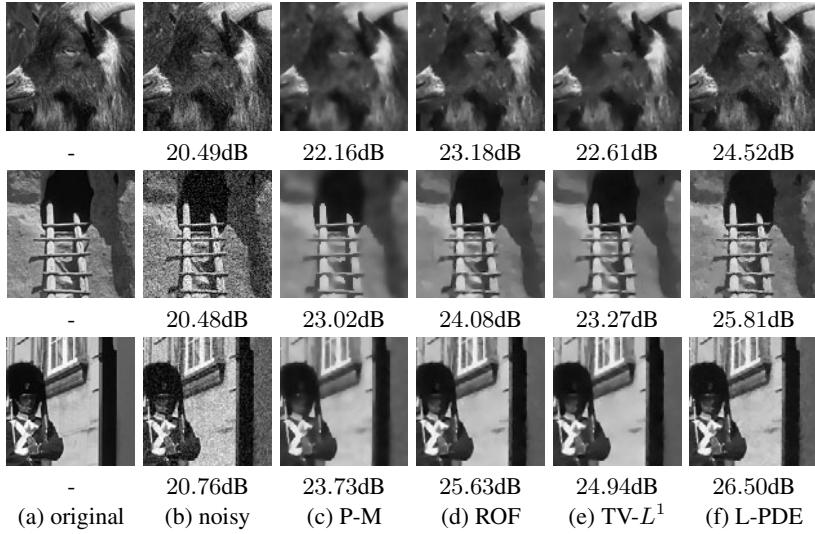
For the image denoising problem, we compare our learnt PDE to the state-of-the-art PDE denoising methods, P-M [5], ROF [9] and TV- $L^1$  [22], on images with both synthetic and real noise. For each experiment, 6 noisy images and their ground truths are randomly chosen to train the coefficients in the L-PDE, and the remaining images are the test images. The parameters in the three compared PDEs are tuned to so that the mean peak signal to noise ratio (PSNR) of all test images are the highest.

**Denoising Images with Synthetic Noise.** We perform two simulation experiments on images with synthetic noise. The images are chosen from the Berkeley image database [23]. There are 86 images in total<sup>6</sup> and the image size is  $321 \times 481$  pixels. For the first experiment, zero-mean Gaussian white noise with  $\sigma = 25$  is added to the images. For the second experiment, a mixture of zero-mean Gaussian white noise ( $\sigma = 50$ ), Poisson noise ( $\lambda$  being the pixel values) and the salt & pepper noise ( $d = 0.1$ ) is added to the images. For both experiments,  $T_f$  is chosen as 2.

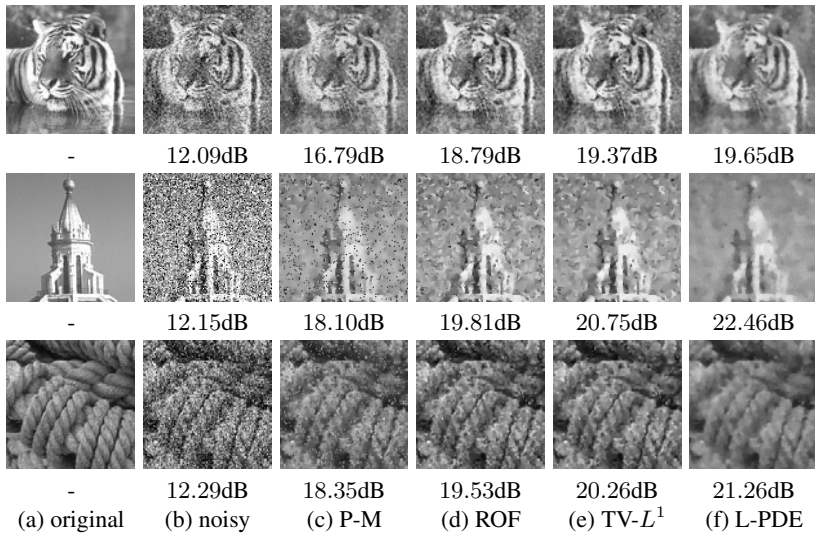
Fig. 1 compares the results of the L-PDE with those of the traditional PDEs on images with Gaussian noise. It shows that the L-PDE preserves details better than the traditional PDEs. Moreover, the L-PDE also achieves higher PSNRs. Fig. 2 shows the comparison of denoising results on mixture noise. One can see that the P-M model cannot remove the salt & pepper noise well. Although ROF and TV- $L^1$  perform better than

---

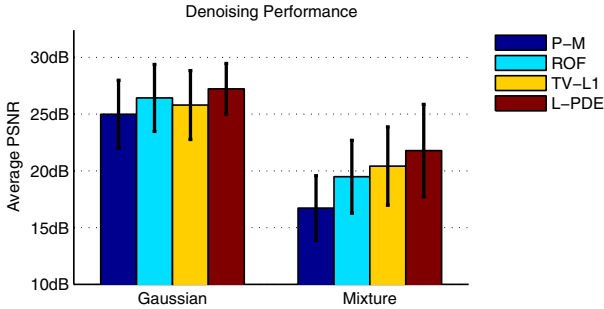
<sup>6</sup> We randomly choose 6 images for training and the remaining 80 images for testing.



**Fig. 1.** The results of denoising images with Gaussian noise. (a) Original noiseless image. (b) Noisy image with additive Gaussian noise ( $\sigma = 25$ ). (c)-(f) Denoised images using the P-M, ROF,  $TV-L^1$ , and our L-PDE models, respectively. The PSNRs are presented below each image.



**Fig. 2.** The results of denoising images with mixture noise. (a) Original noiseless image. (b) Noisy image with mixture noise. (c)-(f) Denoised images using the P-M, ROF,  $TV-L^1$ , and our L-PDE models, respectively. The PSNRs are shown below each image.



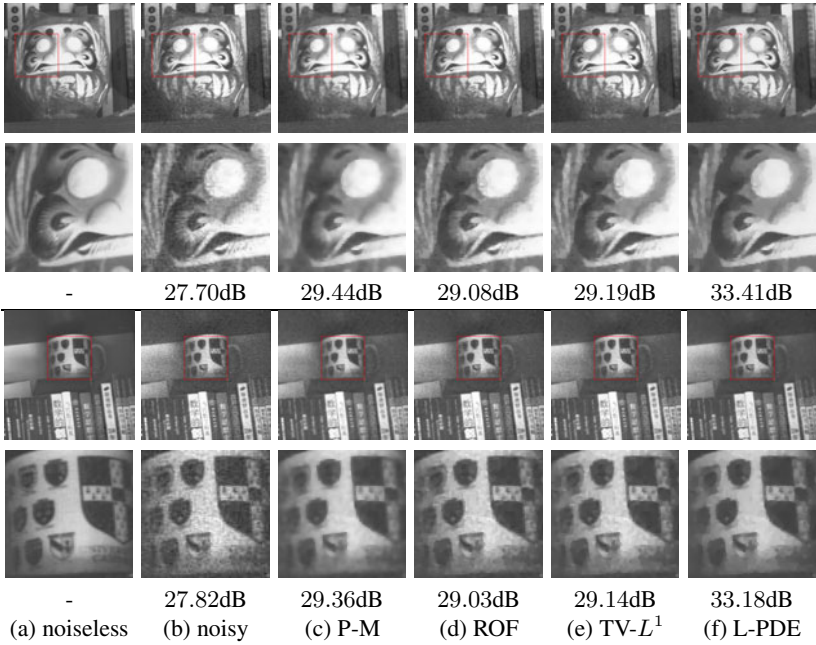
**Fig. 3.** Performance of denoising as measured in PSNR. In each experiment, the average PSNR (colored bar) and standard deviation (thick vertical line) of the denoised images in the test set is shown.

P-M, their denoised images remain noisy. In comparison, our L-PDE suppresses almost all of the noise while preserving the details well.

The quantitative results of the experiments on two kinds of noise are summarized in Fig. 3. One can see that none of the three traditional PDEs can work well on both kinds of noise. On Gaussian noise, ROF outperforms P-M and  $TV-L^1$  and has comparable results with L-PDE, because this model is specifically designed for Gaussian noise. However, ROF does not work well on mixture noise. On mixture noise, the performance of  $TV-L^1$  is better than ROF and P-M. This is because  $TV-L^1$  incorporates a contrast invariant fidelity term, which makes it more adaptive to unknown noise than ROF and P-M. So the performance of the traditional PDEs heavily depends on the test data. In contrast, our L-PDE outperforms all the compared traditional PDEs in both denoising experiments. This is because our L-PDE is data-driven. It learns the form of the PDE from training data to fit the noise, no matter whether the noise distribution is known or unknown.

**Denoising Images with Really Unknown Noise.** To further testify to the data-driven nature of our L-PDE, in this experiment we test on images with really unknown noise. We take 240 images, each with a size  $300 \times 300$  pixels, of 8 objects using a Canon 30D digital camera, setting its ISO to 1600. For each object, 30 images are taken without changing the camera settings (by fixing the focus, aperture and exposure time) and without moving the camera position. The mean image of them can be regarded as the noiseless ground truth image. We randomly choose 6 objects. For each object we randomly choose one noisy image. These noisy images and their ground truth images are used to train an L-PDE, where  $T_f$  is set as 1. Then we compare our L-PDE with the traditional PDEs on images of the remaining 2 objects. In Fig. 4, we show the comparison of these results. The zoomed-in regions show that the output of the L-PDE has less severe artifacts and is sharper than that of other algorithms. As shown in Table 4, the PSNRs of our L-PDE are dramatically higher than those of traditional PDEs. This shows that our L-PDE framework can easily adapt to different types of noise and obtain





**Fig. 4.** The results of denoising images with really unknown noise. The second and fourth rows show the corresponding zoomed-in regions in the boxes in the first and third rows, respectively. (a) The estimated noiseless image. (b) Captured noisy image. ((c)-(f) Denoised images using the P-M, ROF,  $TV-L^1$ , and our L-PDE models, respectively. The estimated PSNRs are shown below each image.

**Table 4.** Denoising results (in PSNR, presented in “mean  $\pm$  std-dev dB”) of the images of the remaining two objects, each object having 30 noisy images

Object	Noisy	P-M	ROF	$TV-L^1$	L-PDE
1	$27.97 \pm 0.19\text{dB}$	$29.55 \pm 0.28\text{dB}$	$29.22 \pm 0.26\text{dB}$	$29.34 \pm 0.27\text{dB}$	$33.25 \pm 0.10\text{dB}$
2	$28.01 \pm 0.31\text{dB}$	$29.89 \pm 0.48\text{dB}$	$29.50 \pm 0.44\text{dB}$	$29.63 \pm 0.45\text{dB}$	$33.36 \pm 0.09\text{dB}$

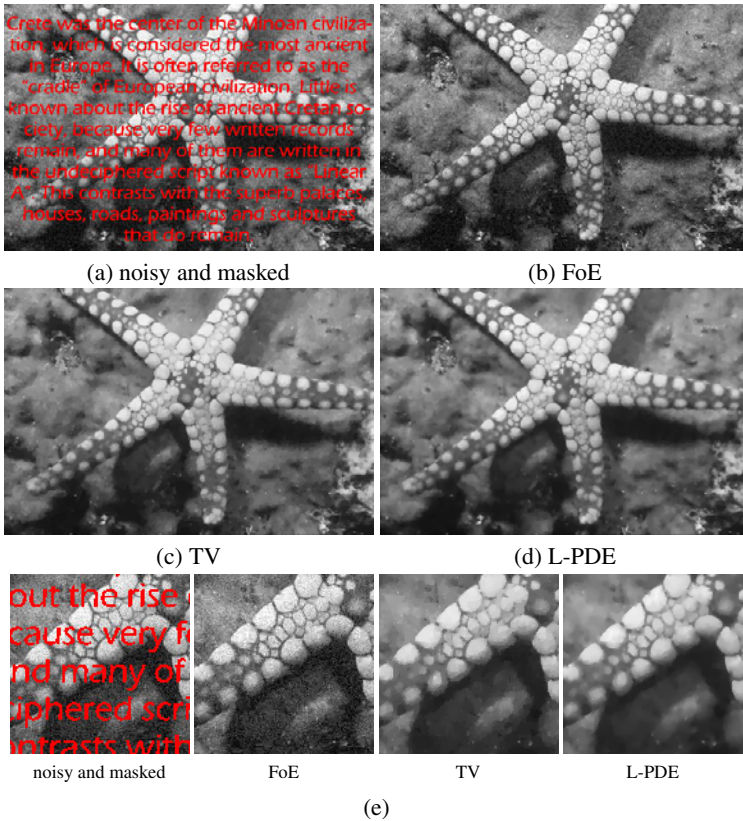
L-PDEs that fit for different types of noise well. In contrast, as the traditional PDEs were designed under specific assumptions on the types of noise, they may not fit for other types of noise as well as our L-PDEs.

## 5.2 Image Inpainting

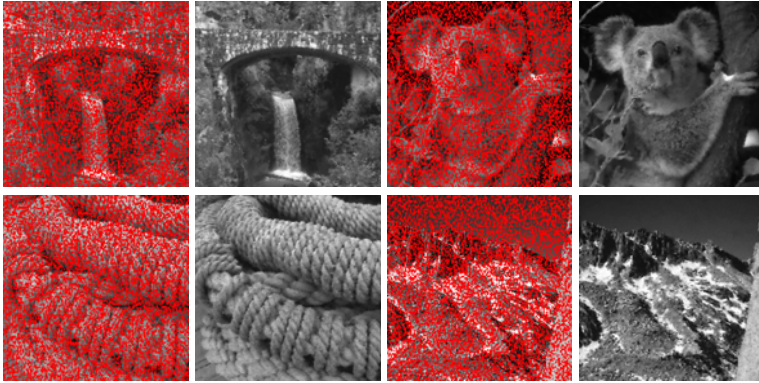
In this subsection, we apply our L-PDE framework to the image inpainting problem. Obviously, to obtain a “good” inpainting result, proper information of the image and the structure of the missing pixels are needed to impose certain priors on the solution. Different from the TV inpainting model [17], which *only* propagates  $\kappa(u)$  to fill in the missing region  $R$ , our L-PDE learns the structure of the missing pixels in  $R$  from the training data and applies both  $\kappa(u)$  and the data-driven term to the test image. As the

data inside the region  $R$  of missing pixels is unavailable, we cannot involve the input image  $f$ , which is  $\text{inv}_0(u)$ , in our L-PDE model. So we limit the coefficient  $a_0(t)$  to be 0 throughout the optimal control process. In this experiment,  $T_f = 4$ .

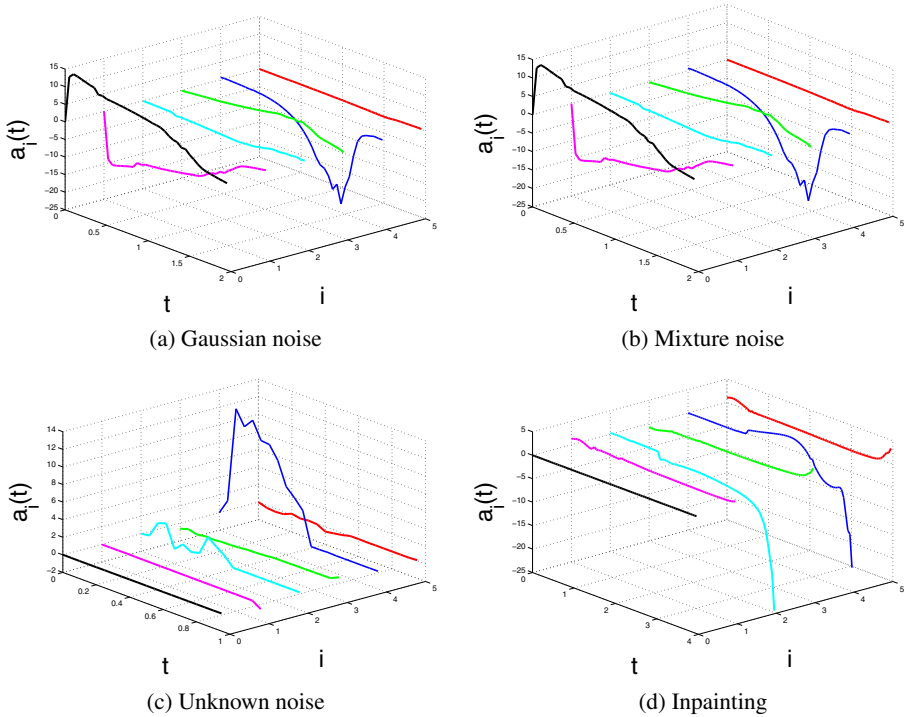
Fig. 5 shows a typical result of noisy image inpainting. We use 6 noisy images (masked by dense text) with their ground truths to train an L-PDE and then apply it to test images. Comparing to the FoE inpainting model [24], which is not a PDE-based method, both TV inpainting [17] and our L-PDE can simultaneously denoise the image and fill in the missing pixels. Moreover, the visual quality and PSNR of our L-PDE are both better than those of TV inpainting [17]. We also apply this L-PDE to other images with purely random masks. Fig. 6 shows that the proposed method also works well.



**Fig. 5.** The results of noisy image inpainting. Gaussian noise with  $\sigma = 15$  is added and then texts are overlaid. PSNRs are computed on the whole image. (a) Noisy image with overlaid text; PSNR = 14.29dB. (b) Inpainting result from FoE; PSNR = 24.42dB. (c) Inpainting result from TV; PSNR = 26.84dB. (d) Inpainting result from L-PDE; PSNR = 27.68dB. (e) Close-up comparison of these algorithms.



**Fig. 6.** The results of purely randomly masked image inpainting (50% pixels are masked), using our L-PDE. The first and the third columns show the masked images. The second and fourth columns show the corresponding inpainted images.



**Fig. 7.** Learnt coefficients  $a_i(t)$ ,  $i = 0, 1, \dots, 5$ , of PDEs for different image restoration problems.

Finally, we show the curves of the learnt coefficients of PDEs for different image restoration problems in Figure 7. Currently we are unable to analyze the obtained PDEs in depth as this work seems to be non-trivial. So we leave the analysis to future work.

## 6 Conclusions and Future Work

In this paper, we have presented a framework of learning PDEs from training data for image restoration. The experiments on natural image denoising and inpainting show that our framework is effective. Compared to the traditional PDEs, our L-PDEs are obtained much more easily. In the future, we would like to improve and enrich our work in the following aspects. First, solve the theoretical issues in our L-PDE model, e.g., the existence and uniqueness of the solution to (1). Second, develop more efficient numerical algorithms to solve our optimal control problem (4). Third, we will also consider incorporating the idea of diffusion tensor [25] and generalizing our framework for vector/matrix/tensor valued images. Finally, we will also apply our framework to more computer vision and image processing problems.

## References

1. Gabor, D.: Information theory in electron microscopy. *Laboratory Investigation* 14, 801–807 (1965)
2. Jain, A.: Partial differential equations and finite-difference methods in image processing, part 1. *Journal of Optimization Theory and Applications* 23, 65–91 (1977)
3. Koenderink, J.: The structure of images. *Biological Cybernetics* 50, 363–370 (1984)
4. Witkin, A.: Scale-space filtering. In: *International Joint Conference on Artificial Intelligence, IJCAI* (1983)
5. Pietro, P., Jitendra, M.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 629–639 (1990)
6. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge detection by non-linear diffusion. *SIAM Journal on Numerical Analysis* 29, 845–866 (1992)
7. Chen, T., Shen, J.: *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM Publisher, Philadelphia (2005)
8. Tikhonov, A., Arsenin, V.: *Solutions of ill-posed problems*. Halsted Press (1977)
9. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* 60, 259–268 (1992)
10. Osher, S., Rudin, L.: Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis* 27, 919–940 (1990)
11. Olver, P.: *Applications of Lie groups to differential equations*. Springer, Heidelberg (1993)
12. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: *Optimization with PDE constraints*. Springer, Heidelberg (2009)
13. Papadakis, N., Corpetti, T., Memin, E.: Dynamically consistent optical flow estimation. In: *International Conference on Computer Vision, ICCV* (2007)
14. Papadakis, N., Memin, E.: Variational optimal control technique for the tracking of deformable objects. In: *International Conference on Computer Vision, ICCV* (2007)
15. Florack, L., Romeny, B., Koenderink, J., Viergever, M.: Scale and the differential structure of image. *Image and Vision Computing* 10, 376–388 (1992)

16. Lindeberg, T.: Discrete derivative approximations with scale-space properties: a basis for low-level feature extraction. *Journal of Mathematical Imaging and Vision* 3, 349–376 (1993)
17. Chan, T., Shen, J.: Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics* 62, 1019–1043 (2002)
18. Stoer, J., Bulirsch, R.: Introduction to numerical analysis, 2nd edn. Springer, Heidelberg (1998)
19. Lions, J.: Optimal control systems governed by partial differential equations. Springer, Heidelberg (1971)
20. Lin, Z., Zhang, W., Tang, X.: Learning partial differential equations for computer vision. Technical report, Microsoft Research, MSR-TR-2008-189 (2008)
21. Lin, Z., Zhang, W., Tang, X.: Designing partial differential equations for image processing by combining differential invariants. Technical report, Microsoft Research, MSR-TR-2009-192 (2009)
22. Chan, T., Esedoglu, S.: Aspects of total variation regularized  $L^1$  function approximation. *SIAM Journal on Applied Mathematics* 65, 1817–1837 (2005)
23. Martin, D.R., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: International Conference on Computer Vision, ICCV (2001)
24. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2005)
25. Weickert, J., Hagen, H.: Visualization and processing of tensor fields. Springer, Heidelberg (2009)

# Compressive Acquisition of Dynamic Scenes\*

Aswin C. Sankaranarayanan<sup>1</sup>, Pavan K. Turaga<sup>2</sup>,  
Richard G. Baraniuk<sup>1</sup>, and Rama Chellappa<sup>2</sup>

<sup>1</sup> Rice University, Houston, TX 77005, USA

<sup>2</sup> University of Maryland, College Park, MD 20740, USA

**Abstract.** Compressive sensing (CS) is a new approach for the acquisition and recovery of sparse signals and images that enables sampling rates significantly below the classical Nyquist rate. Despite significant progress in the theory and methods of CS, little headway has been made in compressive video acquisition and recovery. Video CS is complicated by the ephemeral nature of dynamic events, which makes direct extensions of standard CS imaging architectures and signal models infeasible. In this paper, we develop a new framework for video CS for dynamic textured scenes that models the evolution of the scene as a linear dynamical system (LDS). This reduces the video recovery problem to first estimating the model parameters of the LDS from compressive measurements, from which the image frames are then reconstructed. We exploit the low-dimensional dynamic parameters (the state sequence) and high-dimensional static parameters (the observation matrix) of the LDS to devise a novel compressive measurement strategy that measures only the dynamic part of the scene at each instant and accumulates measurements over time to estimate the static parameters. This enables us to considerably lower the compressive measurement rate considerably. We validate our approach with a range of experiments including classification experiments that highlight the effectiveness of the proposed approach.

## 1 Introduction

Recent advances in the field of compressive sensing (CS) [4] have led to the development of imaging devices that sense at measurement rates below than the Nyquist rate. Compressive sensing exploits the property that the sensed signal is often sparse in some transform basis in order to recover it from a small number of linear, random, multiplexed measurements. Robust signal recovery is possible from a number of measurements that is proportional to the sparsity level of the signal, as opposed to its ambient dimensionality. While there has

---

\* This research was partially supported by the Office of Naval Research under the contracts N00014-09-1-1162 and N00014-07-1-0936, the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number W911NF-09-1-0383, and the AFOSR under the contracts FA9550-09-1-0432 and FA9550-07-1-0301. The authors also thanks Prof. Mike Wakin for valuable discussions and Dr. Ashok Veeraghavan for providing high speed video data.

been remarkable progress in CS for static signals such as images, its application to sensing temporal sequences or videos has been rather limited. Yet, video CS makes a compelling application towards dramatically reducing sensing costs. This manifests itself in many ways including alleviating the data deluge problems faced in the processing and storage of videos.

Existing methods for video CS work under the assumption of the availability of multiple measurements at each time instant. To date, such measurements have been obtained using a snapshot imager [20] or by stacking consecutive measurements from a single pixel camera (SPC) [8]. Given such a sequence of compressive measurements, reconstruction of the video has been approached in multiple directions. Wakin et al. [21] use 3D space-time wavelets as the sparsifying basis for recovering videos from snapshots of compressive measurements. Park and Wakin [12] use a coarse-to-fine estimation framework wherein the video, reconstructed at a coarse level, is used to estimate motion vectors that are subsequently used to design dictionaries for reconstruction at a finer level. Vaswani [16] and Vaswani and Lu [17] propose a sequential framework that exploits the similarity of support and the value the signal takes in this support between adjacent frames of a video. All of these algorithms require a large number of measurements at each time instant and, in most cases, the number of measurements is proportional to the sparsity of an individual frame. This is unsatisfactory as at this compression ratio it is possible to stably reconstruct the individual frames by themselves.

Video CS stands to benefit immensely with the use of strong models characterizing the signals. Park and Wakin [12] use MPEG-like block-matching to improve sparsity of the signal by tuning a wavelet. Veeraraghavan et al. [18] propose a compressive sensing framework of periodic scenes using coded strobing techniques. In this paper, we explore the use of predictive/generative signal models for video CS that are characterized by static parameters. Predictive modeling provides a prior for the evolution of the video in both forward and reverse time. By relating video frames over small durations, predictive modeling helps to reduce the number of measurements required at a given time instant. Models that are largely characterized by static parameters help in eliminating problems arising from the ephemeral nature of dynamic events. Under such a model, measurements taken at *all* time instants contribute towards estimation of the static parameters. At each time instant, it is only required to sense at the rate sufficient to acquire the dynamic component of the scene, which could be significantly lower than the sparsity of an individual frame of the video. One dynamic scene model that exhibits predictive modeling as well as high-dimensional static parameters is the linear dynamical system (LDS). In this paper, we develop methods for the CS of dynamic scenes modeled as LDS motivated, in part, by the extensive use of such models in characterizing dynamic textures [5,7,14], matching shape sequences [19], and activity modeling and video clustering [15].

In particular, the paper makes the following contributions. We propose a framework called *CS-LDS* for video acquisition using a LDS model coupled with sparse priors for the parameters of the LDS model. The core of the proposed framework is a two-step measurement strategy that enables the recovery of LDS

parameters directly from compressive measurements. We solve for the parameters of the LDS using an efficient recovery algorithm that exploits structured sparsity patterns in the observation matrix. Finally, we demonstrate stable recovery of dynamic textures at very low measurement rates.

## 2 Background and Prior Work

**Compressive sensing:** Consider a signal  $\mathbf{y} \in \mathbb{R}^N$ , which is  $K$ -sparse in an orthonormal basis  $\Psi$ ; that is,  $\mathbf{s} \in \mathbb{R}^N$ , defined as  $\mathbf{s} = \Psi^T \mathbf{y}$ , has at most  $K$  non-zero components. Compressive sensing [4,6] deals with the recovery of  $\mathbf{y}$  from undersampled linear measurements of the form  $\mathbf{z} = \Phi \mathbf{y} = \Phi \Psi \mathbf{s}$ , where  $\Phi \in \mathbb{R}^{M \times N}$  is the measurement matrix. For  $M < N$ , estimating  $\mathbf{y}$  from the measurements  $\mathbf{z}$  is an ill-conditioned problem. Exploiting the sparsity of  $\mathbf{s}$ , CS states that the signal  $\mathbf{y}$  can be recovered exactly from  $M = O(K \log(N/K))$  measurements provided the matrix  $\Phi \Psi$  satisfies the so-called *restricted isometry property* (RIP) [1].

In practical scenarios with noise, the signal  $\mathbf{s}$  (or equivalently,  $\mathbf{y}$ ) can be recovered from  $\mathbf{z}$  by solving a convex problem of the form

$$\min \|\mathbf{s}\|_1 \text{ subject to } \|\mathbf{z} - \Phi \Psi \mathbf{s}\| \leq \epsilon \quad (1)$$

with  $\epsilon$  a bound on the measurement noise. It can be shown that the solution to (1) is with high probability the  $K$ -sparse solution that we seek. The theoretical guarantees of CS have been extended to *compressible* signals [10]. In a compressible signal, the sorted coefficients of  $\mathbf{s}$  decay rapidly according to a power-law.

There exist a wide range of algorithms that solve (1) under various approximations or reformulations [4,3]. Greedy techniques such as CoSAMP [11] solve (1) efficiently with strong convergence properties and low computational complexity. It is also easy to impose structural constraints such as block sparsity into CoSAMP giving variants such as model-based CoSAMP [2].

**Dynamic textures and linear dynamical systems:** Linear dynamical systems represent a class of parametric models for time-series data, including dynamic textures [7], traffic scenes [5], and human activities [19,15]. Let  $\{\mathbf{y}_t, t = 0, \dots, T\}$  be a sequence of frames indexed by time  $t$ . The LDS model parameterizes the evolution of  $\mathbf{y}_t$  as follows:

$$\mathbf{y}_t = C \mathbf{x}_t + \mathbf{w}_t \quad \mathbf{w}_t \sim N(\mathbf{0}, R), R \in \mathbb{R}^{N \times N} \quad (2)$$

$$\mathbf{x}_{t+1} = A \mathbf{x}_t + \mathbf{v}_t \quad \mathbf{v}_t \sim N(\mathbf{0}, Q), Q \in \mathbb{R}^{d \times d} \quad (3)$$

where  $\mathbf{x}_t \in \mathbb{R}^d$  is the hidden state vector,  $A \in \mathbb{R}^{d \times d}$  the transition matrix, and  $C \in \mathbb{R}^{N \times d}$  is the observation matrix.

Given the observations  $\{\mathbf{y}_t\}$ , the truncated SVD of the matrix  $[\mathbf{y}]_{1:T} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T]$  can be used to estimate both  $C$  and  $A$ . In particular, an estimate



of the observation matrix  $C$  is obtained using the truncated SVD of  $[\mathbf{y}]_{1:T}$ . Note that the choice of  $C$  is unique only up to a  $d \times d$  linear transformation. That is, given  $[\mathbf{y}]_{1:T}$ , we can define  $\hat{C} = UL$ , where  $L$  is an invertible  $d \times d$  matrix. This represents our choice of coordinates in the subspace defined by the columns of  $C$ . This lack of uniqueness leads to structured sparsity patterns which can be exploited in the inference algorithms.

### 3 Compressive Acquisition of Linear Dynamical Systems

For the rest of the paper, we use the following notation. At time  $t$ , the image observation (the  $t$ -th frame of the video) is  $\mathbf{y}_t \in \mathbb{R}^N$  and the hidden state is  $\mathbf{x}_t \in \mathbb{R}^d$  such that  $\mathbf{y}_t = C\mathbf{x}_t$ , where  $C \in \mathbb{R}^{N \times d}$  is the observation matrix. We use  $\mathbf{z}$  to denote compressive measurements and  $\Phi$  and  $\Psi$  to denote the measurement and sparsifying matrices respectively. We use “ $\cdot$ ” subscripts to denote sequences, such as  $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$  and  $[\cdot]_{1:T}$  to denote matrices, such as  $[\mathbf{y}]_{1:T}$  is the  $N \times T$  matrix formed by  $\mathbf{y}_{1:T}$  such that the  $k$ -th column is  $\mathbf{y}_k$ .

One of the key features of an LDS is that the observations  $\mathbf{y}_t$  lie in the subspace spanned by the columns of the matrix  $C$ . The subspace spanned by  $C$  forms a static parameter of the system. Estimating  $C$  and the dynamics encoded in the state sequence  $\mathbf{x}_{1:T}$  is sufficient for reconstructing the video. For most LDSs,  $N \gg d$ , thereby making  $C$  much higher dimensional than the state sequence  $\{\mathbf{x}_t\}$ . In this sense, the LDS models the video using high information rate static parameters (such as  $C$ ) and low information rate dynamic components (such as  $\mathbf{x}_t$ ). This relates to our initial motivation for identifying signal models with parameters that are largely static. The subspace spanned by  $C$  is static, and hence, we can “pool” measurements over time to recover  $C$ .

Further, given that the observations  $\mathbf{y}_t$  are compressible in a wavelet/Fourier basis, we can argue that the columns of  $C$  need to be compressive as well, either in a similar wavelet basis. This is also motivated by the fact that columns of  $C$  encodes the dominant motion in the scene, and for a large set of videos, this is smooth and has sparse representation in a wavelet/DCT basis or in a dictionary learnt from training data. We can exploit this along the lines of the theory of CS. However, note that  $\mathbf{y}_t = C\mathbf{x}_t$  is a bilinear relationship in  $C$  and  $\mathbf{x}_t$  which complicates direct inference of the unknowns. Towards alleviating this non-linearity, we propose a two-step measurement process that allows to estimate the state  $\mathbf{x}_t$  first and subsequently solve for a sparse approximation of  $C$ . We refer to this as the *CS-LDS* framework.

#### 3.1 Outline of the CS-LDS Framework

At each time instant  $t$ , we take two sets of measurements:

$$\mathbf{z}_t = \begin{pmatrix} \tilde{\mathbf{z}}_t \\ \hat{\mathbf{z}}_t \end{pmatrix} = \begin{bmatrix} \tilde{\Phi} \\ \hat{\Phi}_t \end{bmatrix} \mathbf{y}_t = \Phi_t \mathbf{y}_t, \quad (4)$$

where  $\tilde{\mathbf{z}}_t \in \mathbb{R}^{\tilde{M}}$  and  $\hat{\mathbf{z}}_t \in \mathbb{R}^{\hat{M}}$ , such that the total number of measurements at each frame is  $M = \tilde{M} + \hat{M}$ . Consecutive measurements from an SPC [8] can be

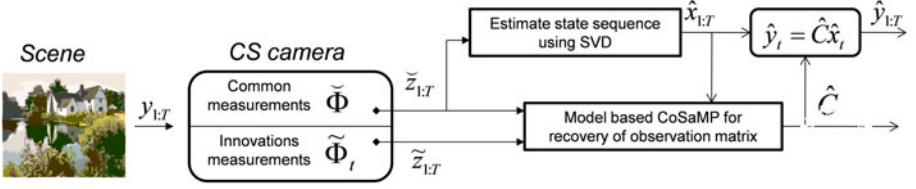


Fig. 1. Block diagram of the CS-LDS framework

aggregated to provide multiple measurements at each  $t$  under the assumption of a quasi-stationary scene. We denote  $\tilde{\mathbf{z}}_t$  as *common* measurements since the corresponding measurement matrix  $\tilde{\Phi}$  is the same at each time instant. We denote  $\tilde{\mathbf{z}}$  as the *innovations* measurements.

The CS-LDS, first, solves for the state sequence  $[\mathbf{x}]_{1:T}$  and subsequently, estimates the observation matrix  $C$ . The common measurements  $[\tilde{\mathbf{z}}]_{1:T}$  are related to the state sequence  $[\mathbf{x}]_{1:T}$  as follows:

$$[\tilde{\mathbf{z}}]_{1:T} = [\tilde{\mathbf{z}}_1 \tilde{\mathbf{z}}_2 \cdots \tilde{\mathbf{z}}_T] = \tilde{\Phi}C [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_T] = \tilde{\Phi}C[\mathbf{x}]_{1:T}. \quad (5)$$

The SVD of  $[\tilde{\mathbf{z}}]_{1:T} = USV^T$  allows us to identify  $[\mathbf{x}]_{1:T}$  up to a linear transformation. In particular, the columns of  $V$  corresponding to the top  $d$  singular values form an estimate of  $[\mathbf{x}]_{1:T}$  up to a  $d \times d$  linear transformation (the ambiguity being the choice of coordinate). When the video sequence is exactly an LDS of  $d$  dimensions, this estimate is exact provided  $\tilde{M} > d$ . The estimate can be very accurate, when the video sequence is approximated by a  $d$ -dimensional subspace as discussed later in this section.

Once we have an estimate of the state sequence, say  $[\hat{\mathbf{x}}]_{1:T}$ , we can obtain  $C$  by solving the following convex problem:

$$(P1) \min \sum_{k=1}^d \|\Psi^T \mathbf{c}_k\|_1, \text{ subject to } \|\mathbf{z}_t - \Phi_t C \hat{\mathbf{x}}_t\|_2 \leq \epsilon, \forall t \quad (6)$$

where  $\mathbf{c}_k$  is the  $k$ -th column of the matrix  $C$ , and  $\Psi$  is a sparsifying basis for the columns of  $C$ . In Section 3.3, we show that the specifics of our measurements induce a structured sparsity in the columns of  $C$ , and this naturally leads to an efficient greedy solution.

To summarize (see Figure 1), the design of the measurement matrix as in (4) enables the estimation of the state sequence using just the common measurements, and subsequently solving for  $C$  using the diversity present in the innovations measurements  $[\tilde{\mathbf{z}}]_t$ .

### 3.2 Random Projections of LDS Data

As mentioned earlier, when  $[\mathbf{y}]_{1:T}$  lies exactly in the (column) span of the matrix  $C$ , then  $[\tilde{\mathbf{z}}]_{1:T}$  lies in the span of  $\tilde{\Phi}C$ . Hence, the SVD of  $[\tilde{\mathbf{z}}]_{1:T}$  can be used to recover the state sequence up to a linear transformation, provided  $\tilde{M} \geq d$

$$[\tilde{\mathbf{z}}]_{1:T} = USV^T, \quad [\hat{\mathbf{x}}]_{1:T} = S_d V_d^T \quad (7)$$

where  $S_d$  is the  $d \times d$  principal sub-matrix of  $S$  and  $V_d$  is the  $T \times d$  matrix formed by columns of  $V$  corresponding to the largest  $d$  singular values. In practice, the observations  $\mathbf{y}_t$  lie close to the subspace spanned by  $C$  such that projection of onto  $C$  makes for a highly accurate approximation of  $\mathbf{y}_t$ . In such a case, the estimate of the state sequence from the SVD of  $[\tilde{\mathbf{z}}]_{1:T}$  is accurate only when the observations  $\mathbf{y}_t$  are compressible [9]. In our case, this is equivalent to imposing a power-law decay on the singular values. Figure 2 shows the accuracy of the approximation of the estimated state sequence for various values of  $\tilde{M}$ . This suggests that, in practice,  $\mathbf{x}_t$  can be reliably estimated with  $\tilde{M} \propto d$ .

### 3.3 Structured Sparsity and Recovery with Modified CoSAMP

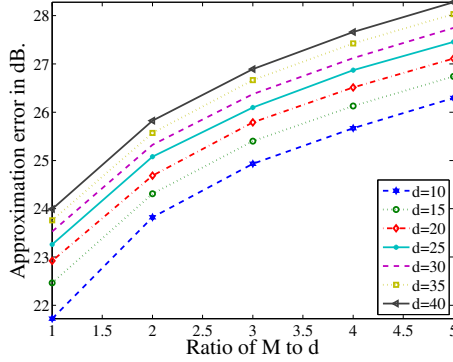
The SVD of the common compressive measurements  $\tilde{\mathbf{z}}_t$  introduces an ambiguity in the estimates of the state sequence in the form of  $[\hat{\mathbf{x}}]_{1:T} \approx L^{-1}[\mathbf{x}]_{1:T}$ , where  $L$  is an invertible  $d \times d$  matrix. Solving (P1) using this estimate will, at best, lead to an estimate  $\hat{C} = CL$  satisfying  $\mathbf{z}_t = \Phi_t \hat{C} \hat{\mathbf{x}}_t$ . This ambiguity introduces additional concerns in the estimation of  $C$ . Suppose the columns of  $C$  are  $K$ -sparse (equivalently, compressible for a certain value of  $K$ ) each in  $\Psi$  with support  $\mathcal{S}_k$  for the  $k$ -th column. Then, the columns of  $CL$  are potentially  $dK$ -sparse with identical supports  $\mathcal{S} = \bigcup_k \mathcal{S}_k$ . The support is exactly  $dK$ -sparse when the  $\mathcal{S}_k$  are disjoint and  $L$  is dense. At first glance, this seems to be a significant drawback, since the overall sparsity of  $\hat{C}$  has increased to  $d^2K$ . However, this apparent increase in sparsity is alleviated by the columns having identical supports. The property of identical supports on the columns of  $CL$  can be exploited to solve (P1) very efficiently using greedy methods.

Given the state sequence, we use a modified CoSAMP algorithm to estimate  $C$ . The modification exploits the structured sparsity induced by the columns of  $C$  having identical support. In this regard, the resulting algorithm is a particular instance of the model-based CoSAMP algorithm [2]. One of the key properties of model-based CoSAMP is that stable signal recovery requires only a number of measurements that is proportional to the model-sparsity of the signal, which in our case is equal to  $dK$ . Hence, we can recover the observation matrix from  $O(dK \log(Nd))$  measurements [2]. Figure 3 summarizes the model-based CoSAMP algorithm used for recovering the observation matrix  $C$ .

### 3.4 Performance and Measurement Rate

For a stable recovery of the observation matrix  $C$ , we need in total  $O(dK \log(Nd))$  measurements. In addition to this, for recovering the state sequence, we need a number of common measurements proportional to the dimensionality of the state vectors

$$MT \propto dK \log(Nd), \quad \tilde{M} \propto d. \quad (8)$$



**Fig. 2.** Average error in estimating the state sequence from common measurements for various values of state dimension  $d$  and the ratio  $\widehat{M}/d$ . Statistics were computed using 114 videos of 250 frames taken from the DynTex database [13].

$$\widehat{C} = \text{CoSaMP\_Model\_Sparsity}(\Psi, K, \mathbf{z}_t, \widehat{\mathbf{x}}_t, \Phi_t, t = 1, \dots, T)$$

**Notation:**

$\text{supp}(\text{vec}; K)$  returns the support of  $K$  largest elements of  $\text{vec}$

$A_{|\Omega, \cdot}$  represents the submatrix of  $A$  with rows indexed by  $\Omega$  and all columns.

$A_{\cdot, |\Omega}$  represents the submatrix of  $A$  with columns indexed by  $\Omega$  and all rows.

$$\forall t, \Theta_t \leftarrow \Phi_t \Psi$$

$$\forall t, \mathbf{v}_t \leftarrow \mathbf{0} \in \mathbb{R}^M$$

$$\Omega_{\text{old}} \leftarrow \phi$$

While (stopping conditions are not met)

$$R = \sum_t \Theta_t^T \mathbf{v}_t \widehat{\mathbf{x}}_t^T \quad (R \in \mathbb{R}^{N \times d})$$

$$k \in [1, \dots, N], \mathbf{r}(k) = \sum_{i=1}^d R^2(k, i) \quad (\mathbf{r} \in \mathbb{R}^N)$$

$$\Omega \leftarrow \Omega_{\text{old}} \cup \text{supp}(\mathbf{r}; 2K)$$

Find  $A \in \mathbb{R}^{|\Omega| \times d}$  that minimizes  $\sum_t \|\mathbf{z}_t - (\Theta_t)_{|\Omega, \cdot} A \widehat{\mathbf{x}}_t\|_2$

$$B_{|\Omega, \cdot} \leftarrow A$$

$$B_{|\Omega^c, \cdot} \leftarrow 0$$

$$k \in [1, \dots, N], \mathbf{b}(k) = \sum_{i=1}^d B^2(k, i) \quad (\mathbf{b} \in \mathbb{R}^N)$$

$$\Omega \leftarrow \text{supp}(\mathbf{b}; K)$$

$$S_{|\Omega, \cdot} \leftarrow B_{|\Omega, \cdot} \quad S_{|\Omega^c, \cdot} \leftarrow 0$$

$$\Omega_{\text{old}} \leftarrow \Omega$$

$$\widehat{C} \leftarrow \Psi B$$

$$\forall t, \mathbf{v}_t \leftarrow \mathbf{z}_t - \Theta_t S \widehat{\mathbf{x}}_t$$

**Fig. 3.** Pseudo-code of the model-based CoSAMP algorithm for CS-LDS

Compared to Nyquist sampling, we obtain a measurement rate ( $M/N$ ) given by

$$\frac{M}{N} \propto \frac{dK \log(Nd)}{NT}. \quad (9)$$

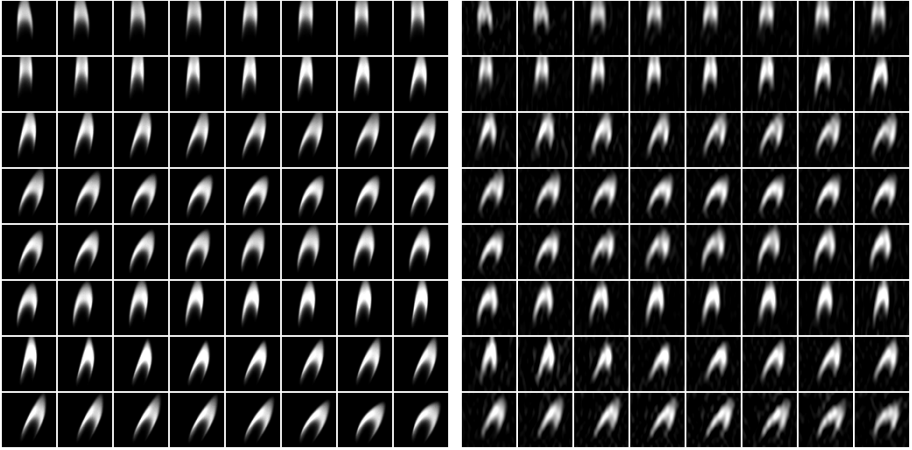
This indicates extremely favorable operating scenarios for the CS-LDS framework, especially when  $T$  is large (as in a high frame rate capture). Consider a segment of a video of *fixed* duration observed at various sampling rates. The effective number of frames,  $T$ , changes with the sampling rate,  $f_s$  (in frames per second), as  $T \propto f_s$ . However, the complexity of the video measured using the state space dimension  $d$  does not change. Hence, as the sampling rate  $f_s$  increases,  $M$  can be decreased while keeping  $Mf_s$  constant. This will ensure that (8) is satisfied, enabling a stable recovery of  $C$ . This suggests that as the sampling rate  $f_s$  increases our measurement rate decreases, a very desirable property for high-speed imaging.

### 3.5 Extensions

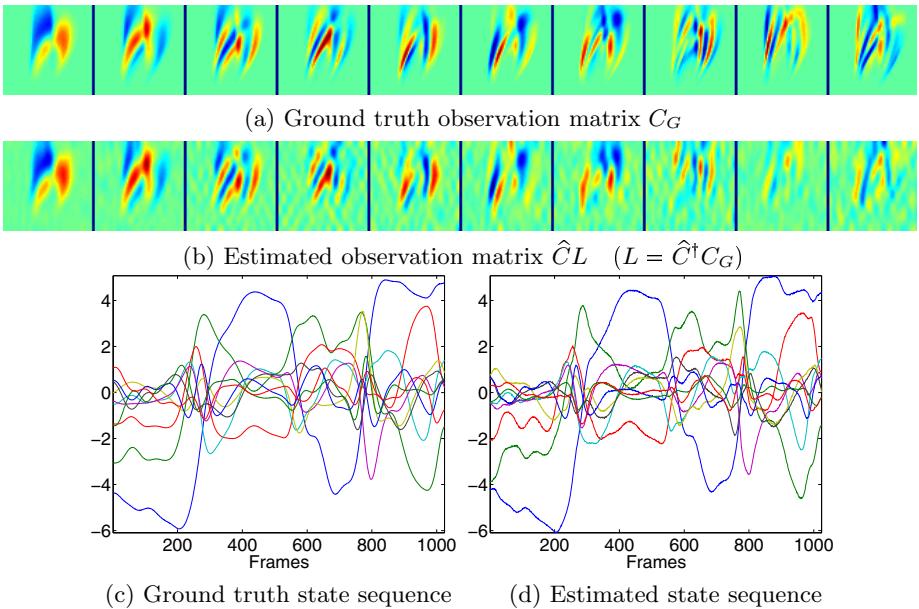
**Mean + LDS:** In many instances, a dynamical scene is modeled better as an LDS over a static background, that is,  $\mathbf{y}_t = C\mathbf{x}_t + \mu$ . This can be handled with two minimal modifications to the algorithm described above. First, the state sequence  $[\hat{\mathbf{x}}]_{1:T}$  is obtained by performing SVD on the matrix  $[\tilde{\mathbf{z}}]_{1:T}$  modified such that the each row sums to zero. This works under the assumption that the sample mean of  $\tilde{\mathbf{z}}_{1:T}$  is equal to  $\tilde{\Phi}\mu$ , the compressive measurement of  $\mu$ . Second, we use model-based CoSAMP to estimate both  $C$  and  $\mu$  simultaneously. However, only the columns of  $C$  enjoy the structured sparsity model. The support of  $\mu$  is not constrained to be similar to that of  $C$ .

## 4 Experimental Validation

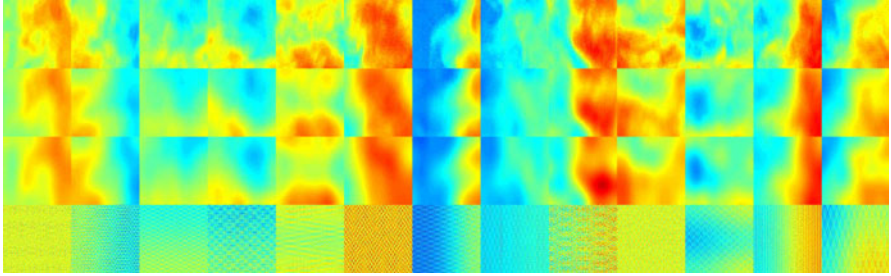
We present a range of experiments validating various aspects of the CS-LDS framework. Our test dataset comprises of videos from DynTex [13] and data we collected using high speed cameras. For most experiments, we chose  $\tilde{M} = 2d$ , with  $d$  and  $K$  chosen appropriately. We used the mean+LDS model for all the experiments with the 2D DCT as the sparsifying basis for the columns of  $C$  as well as the mean. Finally, the entries of the measurement matrix were sampled from iid standard Gaussian distribution. We compare against *frame-by-frame* CS where each frame of the video is recovered separately using conventional CS techniques. We use the term *oracle LDS* for parameters and video reconstruction obtained by operating on the original data itself. The oracle LDS estimates the parameters using a rank- $d$  approximation to the ground truth data. The reconstruction SNR of the oracle LDS gives an upper bound on achievable SNR. Finally, the ambiguity in observation matrix (due to non-uniqueness of the SVD based factorization) as estimated by oracle LDS and CS-LDS is resolved for visual comparison in Figures 5 and 6.



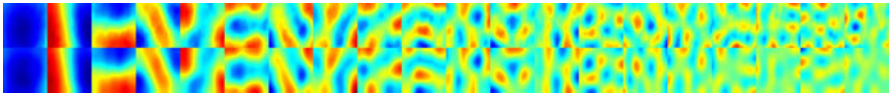
**Fig. 4.** Reconstruction of  $T = 1024$  frames of a scene of resolution  $N = 64 \times 64$  pixels shown as a mosaic. The original data was collected using a high speed camera operating at 1000 fps. Compressive measurements were obtained with  $\bar{M} = 30$  and  $\bar{M} = 20$ , thereby giving a measurement rate  $M/N = 1.2\%$ . Reconstruction was performed using an LDS with  $d = 15$  and  $K = 150$ . Shown above are 64 uniformly sampled frames from the ground truth (left) and the reconstruction (right).



**Fig. 5.** Ground truth and estimated parameters corresponding to Figure 4. Shown are the top 10 columns of the observation matrix and state sequences. Matlab’s “jet” colormap (red= +large and blue= -large) is used in (a) and (b).



(a) Mosaic of frames of a video, with each column a different time instant, and each row a different algorithm. (top row to bottom) ground truth, oracle LDS, CS-LDS, and frame-by-frame CS.



(b) Mosaic of ground truth (top) and estimated (bottom) observation matrix

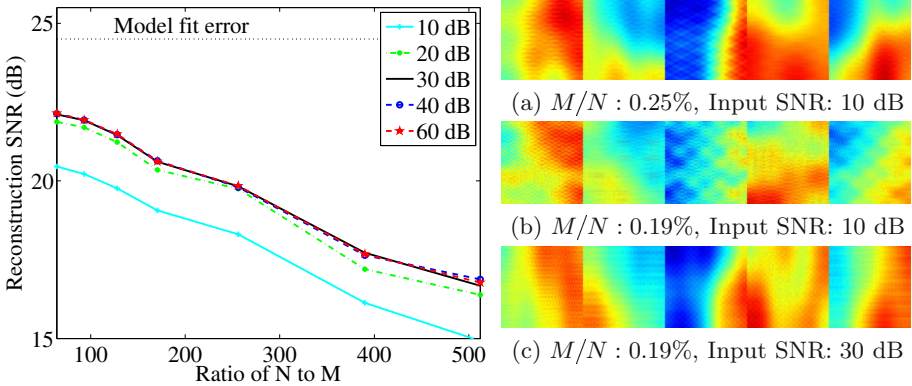
**Fig. 6.** Reconstruction of a fire texture of length 250 frames and resolution of  $N = 128 \times 128$  pixels. Compressive measurements were obtained at  $\tilde{M} = 30$  and  $\tilde{M} = 40$  measurements per frame, there by giving a measurement rate of 0.42% of Nyquist. Reconstruction was performed with  $d = 20$  and  $K = 30$ . Frames of the videos are shown in false-color for better contrast.

**Reconstruction:** Figure 4 shows reconstruction results from data collected from a high speed camera of a candle flame. Figure 5 shows the estimated observation matrix as well as the state sequence.

Figure 6 shows video reconstruction of a dynamic texture from the DynTex dataset [3]. Reconstruction results are under a measurement rate  $M/N = 1/234$  (about 0.42%), an operating point where a frame-to-frame CS recovery is completely infeasible. However, the dynamic component of the scene is relatively small ( $d = 20$ ) which allows us to recover the video from relatively few measurements. The SNR of the reconstructions shown are as follows: Oracle LDS = 24.97 dB, frame-to-frame CS: 11.75 dB and CS-LDS: 22.08 dB.

**Performance with measurement noise:** It is worth noting that the video sequences used in the experiments have moderate model fit error at a given value of  $d$ . The columns of  $C$  with larger singular values are, inherently, better conditioned to deal with this model error. The columns corresponding to the smaller singular values are invariably estimated at higher error. This is reflected in the estimates of the  $C$  matrix in Figures 5 and 6.

Figure 7 shows the performance of the recovery algorithm for various levels of measurement noise. The effect of the measurement noise on the reconstructions is perceived only at much lower SNR. This is, in part, due to the model fit error dominating the performance of the algorithm when the measurement noise SNR is very high. As the measurement SNR drops significantly below the model fit error, predictably, it starts influencing the reconstructions more. This provides a certain amount of flexibility in the design of potential CS-LDS cameras especially in scenarios where we are not primarily interested in visualization of the sensed video.

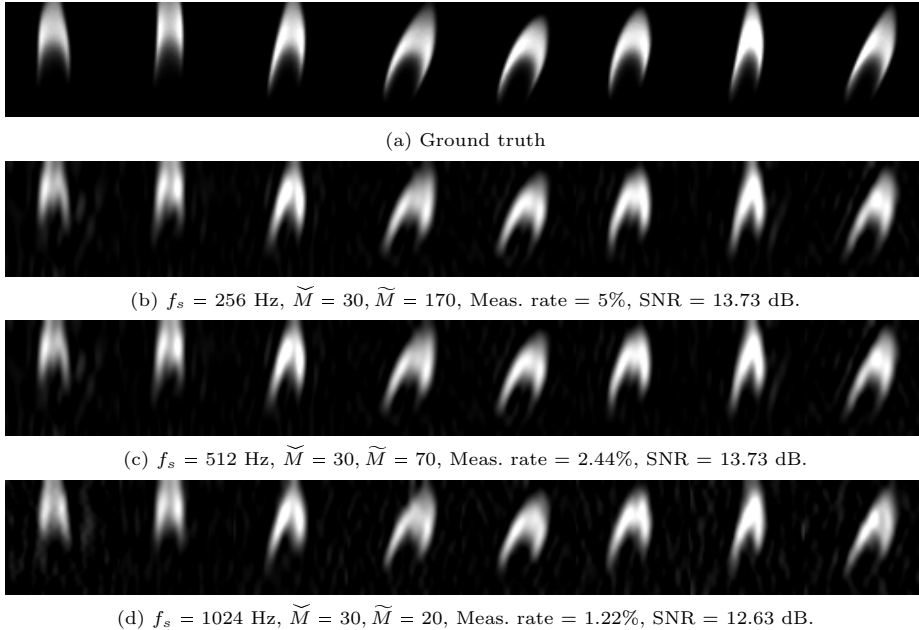


**Fig. 7.** Resilience of the CS-LDS framework to measurement noise. (Left) Reconstruction SNR as a function of measurement rates and input SNR levels computed using 32 Monte-Carlo simulations. The “black-dotted” line shows the reconstruction SNR for an  $d = 20$  oracle LDS. (Right) Snapshots at various operating points. The dynamic texture of Figure 6 was used for this result.

**Sampling rate:** Figure 8 shows reconstruction plots of the candle sequence (of Figure 4) for 1 second of video at various sampling rates. We use (9) to predict the required measurement rates at various sampling rates to maintain a constant reconstruction SNR. As expected, the reconstruction SNR remains the same, while the measurement rate decreases significantly with a linear increase in the sampling rate. This makes the CS-LDS framework extremely promising for high speed capture applications. In contrast, most existing video CS algorithms have measurement rates that, at best, remain constant as the sampling rate increases.

**Application to scene classification:** In this experiment, we study feasibility of classification problems on the videos sensed and reconstructed under the CS-LDS framework. We consider the UCSD traffic database used in 5. The dataset consists of 254 videos of length 50 frames capturing traffic of three types: light, moderate, heavy. Figure 9 shows reconstruction results on a traffic sequence from the dataset. We performed a classification experiment of the videos into these three categories. There are 4 different train-test scenarios provided with the dataset. Classification is performed using the subspace-angles based metric with a nearest-neighbor classifier on the LDS parameters 14. The experiment was performed using the parameters estimated directly without reconstructing the frames. For comparison, we also perform the same experiments with fitting the LDS model on the original frames (oracle LDS). Table 1 shows classification results. We see that we obtain comparable classification performance using the proposed CS-LDS recovery algorithm to the oracle LDS. This suggests that the CS-LDS camera is extremely useful in a wide range of applications not tied to video recovery.





**Fig. 8.** As the sampling frequency  $f_s$  increases, we maintain the same reconstruction capabilities for significantly lesser number of measurements. Shown are reconstructions for  $N = 64 \times 64$  and various sampling frequencies, achieved measurement rates, and reconstruction SNRs.

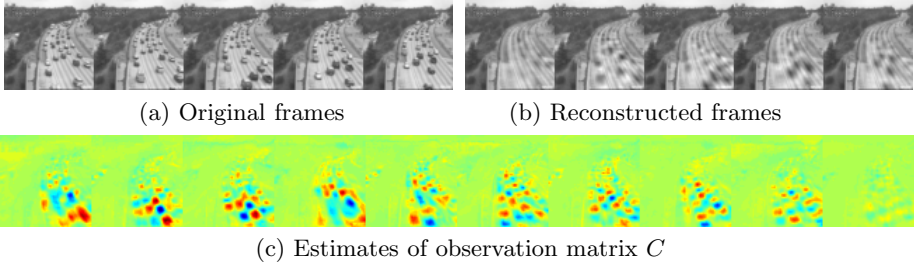
**Table 1.** Classification results (in %) on the traffic databases for two different values of state space dimension  $d$ . Results are over a database of 254 videos, each of length 50 frames at a resolution of  $64 \times 64$  pixels under a measurement rate of 4%.

(a) $d = 10$					(b) $d = 5$				
	Expt 1	Expt 2	Expt 3	Expt 4		Expt 1	Expt 2	Expt 3	Expt 4
Oracle LDS	85.71	85.93	87.5	92.06	Oracle LDS	77.77	82.81	92.18	80.95
CS-LDS	84.12	87.5	89.06	85.71	CS-LDS	85.71	73.43	78.1	76.1

## 5 Discussion

In this paper, we proposed a framework for the compressive acquisition of dynamic scenes modeled as LDSs. We show that the strong scene model for the video enables stable reconstructions at very low measurement rates. In particular, this emphasizes the power of video models that are predictive as well as static.

**Extensions of the CS-LDS framework:** The CS-LDS algorithm proposed in this paper requires, at best,  $O(d)$  measurements per time instant. This roughly corresponds to the number of degrees of freedom in the dynamics of the video un-



**Fig. 9.** Reconstructions of a traffic scene of  $N = 64 \times 64$  pixels at a measurement rate 4%, with  $d = 15$  and  $K = 40$ . The quality of reconstruction and LDS parameters is sufficient for capturing the flow of traffic.

der a  $d$ -dimensional LDS model. However, the state transition model of the LDS further constrains the dynamics by providing a model for the evolution of the signal. Incorporating this might help in reducing the number of measurements required at each time instant. Another direction for future research is in fast recovery algorithms that operate at multiple spatio-temporal scales, exploiting the fact that a global LDS model induces a local LDS model as well. Finally, much of the proposed algorithm relies on sparsity of the observation matrix  $C$ . Wavelets and Fourier (DCT) bases do not sparsify videos where the motion is localized in space. This suggests the use of dyadic partition methods such as platelets [22], which have been shown to have success in modeling bounded shapes.

**Newer models for video CS:** While the CS-LDS framework makes a compelling case study of LDSs for video CS, its applicability to an arbitrary video is limited. The LDS model is well-matched to a large class of dynamic textures such as flames, water, traffic etc. but does not extend to simple non-stationary scenes such as people walking. The importance of video models for CS motivates the search for models that are more general than LDS. In this regard, a promising line of future research is to leverage our new understanding of video models for compression algorithm-based CS recovery.

## References

1. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. *Constructive Approximation* 28(3), 253–263 (2008)
2. Baraniuk, R., Cevher, V., Duarte, M., Hegde, C.: Model-Based Compressive Sensing. *IEEE Transactions on Information Theory* 56(4), 1982–2001 (2010)
3. van den Berg, E., Friedlander, M.P.: Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing* 31(2), 890–912 (2008)
4. Candès, E., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory* 52(2), 489–509 (2006)

5. Chan, A.B., Vasconcelos, N.: Probabilistic kernels for the classification of autoregressive visual processes. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 846–851 (2005)
6. Donoho, D.: Compressed sensing. *IEEE Transactions on Information Theory* 52(4), 1289–1306 (2006)
7. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. *International Journal of Computer Vision* 51(2), 91–109 (2003)
8. Duarte, M., Davenport, M., Takhar, D., Laska, J., Sun, T., Kelly, K., Baraniuk, R.: Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine* 25(2), 83–91 (2008)
9. Fowler, J.: Compressive-projection principal component analysis. *IEEE Transactions on Image Processing* 18(10) (October 2009)
10. Haupt, J., Nowak, R.: Signal reconstruction from noisy random projections. *IEEE Transactions on Information Theory* 52(9), 4036–4048 (2006)
11. Needell, D., Tropp, J.: CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis* 26(3), 301–321 (2009)
12. Park, J., Wakin, M.: A multiscale framework for compressive sensing of video. In: *Picture Coding Symposium*, pp. 197–200 (May 2009)
13. Péteri, R., Fazekas, S., Huiskes, M.: DynTex: A Comprehensive Database of Dynamic Textures (to appear, 2010), <http://projects.cwi.nl/dyntex/>
14. Saisan, P., Doretto, G., Wu, Y., Soatto, S.: Dynamic texture recognition. In: *CVPR*. vol. 2, pp. 58–63 (December 2001)
15. Turaga, P., Veeraraghavan, A., Chellappa, R.: Unsupervised view and rate invariant clustering of video sequences. *CVIU* 113(3), 353–371 (2009)
16. Vaswani, N.: Kalman filtered compressed sensing. In: *ICIP* (2008)
17. Vaswani, N., Lu, W.: Modified-CS: Modifying compressive sensing for problems with partially known support. In: *Intl. Symposium on Information Theory* (2009)
18. Veeraraghavan, A., Reddy, D., Raskar, R.: Coded strobing photography: Compressive sensing of high-speed periodic events. *TPAMI* (to appear)
19. Veeraraghavan, A., Roy-Chowdhury, A.K., Chellappa, R.: Matching shape sequences in video with applications in human movement analysis. *TPAMI* 27, 1896–1909 (2005)
20. Wagadarikar, A., John, R., Willett, R., Brady, D.: Single disperser design for coded aperture snapshot spectral imaging. *Applied Optics* 47(10), 44–51 (2008)
21. Wakin, M., Laska, J., Duarte, M., Baron, D., Sarvotham, S., Takhar, D., Kelly, K., Baraniuk, R.: Compressive imaging for video representation and coding. In: *Picture Coding Symposium* (April 2006)
22. Willett, R., Nowak, R.: Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Transactions on Medical Imaging* 22(3), 332–350 (2003)

# Scene Carving: Scene Consistent Image Retargeting

Alex Mansfield<sup>1</sup>, Peter Gehler<sup>1</sup>, Luc Van Gool<sup>1,2</sup>, and Carsten Rother<sup>3</sup>

<sup>1</sup> Computer Vision Laboratory, ETH Zürich, Switzerland

<sup>2</sup> ESAT-PSI, KU Leuven, Belgium

<sup>3</sup> Microsoft Research Ltd, Cambridge, UK

{mansfield,pgehler,vangool}@vision.ee.ethz.ch, carrot@microsoft.com

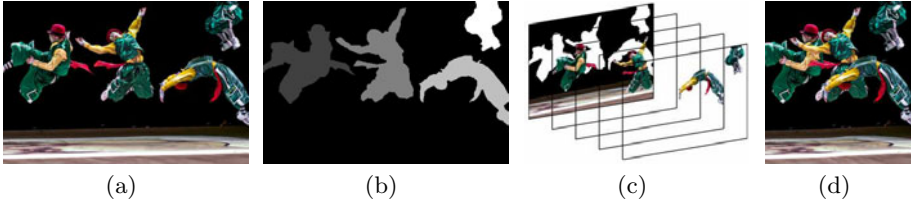
**Abstract.** Image retargeting algorithms often create visually disturbing distortion. We introduce the property of scene consistency, which is held by images which contain no object distortion and have the correct object depth ordering. We present two new image retargeting algorithms that preserve scene consistency. These algorithms make use of a user-provided relative depth map, which can be created easily using a simple GrabCut-style interface. Our algorithms generalize seam carving. We decompose the image retargeting procedure into (a) removing image content with minimal distortion and (b) re-arrangement of known objects within the scene to maximize their visibility. Our algorithms optimize objectives (a) and (b) jointly. However, they differ considerably in how they achieve this. We discuss this in detail and present examples illustrating the rationale of preserving scene consistency in retargeting.

## 1 Introduction

The increasing diversity of modern displays calls for methods able to transform images so as to best exploit the display form factor. Such *media retargeting* has received much attention lately [1, 2, 4–8, 10, 11, 14, 18, 19, 22, 23]. Recent success can be attributed to two developments: firstly, the use of “content-aware” algorithms with more accurate image models; secondly, the formulation of the problem as a graph labelling problem, for which efficient solvers exist [3, 21].

Most existing approaches are fully automatic, using low level visual saliency to determine image region importance. These suffer problems with structured objects, which low level saliency is not able to detect. However, we assume that a *relative depth map* is available, provided by the user. By a relative depth map, we refer to object segmentations with a depth order label, as illustrated in Fig. 1(b).

Given this depth map, our novel retargeting algorithms are capable of retargeting such that objects are protected (i.e. not distorted) and maintain their correct depth ordering. We term this condition *scene consistency*. We extend the well-known seam carving algorithm [1] to achieve this. To the best of our knowledge, these are the first retargeting algorithms that are able to re-arrange objects such that object occlusions are created, as illustrated in Fig. 1.



**Fig. 1.** For image (a) with relative depth map (b), illustrated in 3D in (c), we produce the scene consistent retargeted image (d) by the new *scene carving* algorithm

We acknowledge that assuming our additional input is a strong assumption, but the improvement in the output can make its acquisition worthwhile. Furthermore, recent developments allow the input to be acquired relatively easily. Firstly, efficient interactive user interfaces are now available for such annotation. In our work we make use of an interface employing the GrabCut algorithm [13], with which all our depth maps were created within a few minutes. Secondly, recent work [9, 16] has begun to succeed in detecting occlusion boundaries and acquiring 3D models from single images. These techniques could be used in automating, at least partially, the annotation process. Thirdly, commercial stereo cameras are hitting the market<sup>1</sup>. With state-of-the-art stereo depth estimation techniques [17], this technology may allow complete automation of this process.

In the next section we discuss related work on image retargeting. In Sect. 3 we discuss the properties of scene consistent retargeted images. Sections 4 and 5 contain the proposed algorithms. Real world examples are shown in Sect. 6 and we conclude with a discussion on future work in Sect. 7.

## 2 Related Work

There exists a large body of literature on media retargeting. In this section we discuss work which is most relevant to ours. Please note that we focus on image retargeting, although many algorithms have been extended to video.

**On Retargeting.** Two main strategies exist for image retargeting: minimizing applied distortion or maximizing similarity between the input and output images.

Arguably the simplest retargeting methods are cropping and scaling. These methods usually are not content aware and tend to give inferior results to algorithms that are. Some work exists on content-aware scaling and cropping [15, 19, 20] but these methods alone have limited ability to retain content or can cause distortions such that interesting parts of the image are no longer clearly visible.

Seam carving [1, 14] has received a lot of attention due to its elegance. It iteratively removes connected paths of pixels so as to minimize the resulting distortion. It can be thought of as forgetting the input image altogether, as the

<sup>1</sup> E.g. Fuji FinePix 3D W1.

[www.fujifilm.com/products/3d/camera/finepix\\_real3dw1](http://www.fujifilm.com/products/3d/camera/finepix_real3dw1)

distortion it measures is relative only to the previous image. Together with our algorithms and other extensions [7, 8, 15], it falls under the first strategy. We build on it for reasons of speed and because of the ability to explicitly control the modifications of pixels. We discuss this in greater detail in Sect. 4.1.

The second strategy requires a notion of distance between the input and output image. Many have been proposed and used in retargeting, based on e.g. patch colour similarity [2, 15, 19] with dominant colours [5], saliency with image gradients [22] or with face attention [18], and colour and gradient difference [11].

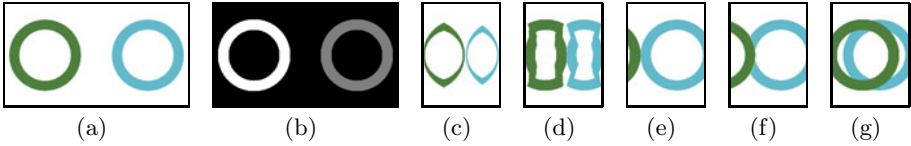
**On Protecting Objects.** Object protection (i.e. non-distortion) is important for the realism of synthesized images. In dynamic video synopsis [12], objects are detected using background subtraction, and protected in the synopsis. In [6] the user is requested to mark parts of the image where shape should be preserved. In [1], users can specify regions to be protected or removed during retargeting.

The method proposed by [18] is closest to our approach with regard to object protection. Importance maps are created automatically, from which important regions are detected. The retargeted output is constructed by removing the important regions, inpainting the resulting holes in the background, rescaling the background, and finally re-inserting and re-arranging the removed regions to create the output. The important regions thus avoid the rescaling, and so are protected. The authors show results which are visually pleasing, but the method relies on the strength of the inpainting algorithm. Also, unlike our methods, it is not able to create consistent object occlusions.

### 3 Scene Consistency

We first introduce the key concept of scene consistency. We model image formation as projection of flat fronto-parallel objects at different depths onto a background plane. An image can be decomposed into such a model as illustrated in Fig. 1(c). A retarget of the image is *scene consistent* if objects (1) are not distorted but kept as in the original image and (2) are placed in their correct depth ordering. We also define the concept of object consistency, which is held by retargets for which property (1) holds, that objects are not distorted.

This concept provides a formalization of scene realism, which we want to maintain during retargeting. To do so requires the model decomposition of the original image, which for a single image can be described simply in terms of a relative depth map, giving object segmentations each with a depth ordering label as illustrated in Fig. 1(b). Object segmentations alone allow scene consistent retargeting, by enforcing no distortion for the objects, but with the depth information, scene consistent occlusions may also be generated. The benefits of scene consistent retargeting are illustrated for a toy image in Fig. 2. Note that we distinguish between occlusions that require reappearance and those that do not, a distinction we find arises in practice. By “reappearance” we refer to pixels previously occluded becoming visible again while iterative retargeting.



**Fig. 2.** Toy image (a) with depth layers (b) is retargeted by seam carving [11] (c), seam carving with object protection (Sect. 4.1 [11]) (d) and (e), seam carving with occlusions (Sect. 4.2) (f) and scene carving (Sect. 5) (g). Our two new algorithms (f, g) may form occlusions: in seam carving with occlusions (f), occlusions that do not require reappearance may be formed (see Sect. 4.3); in scene carving (g), all scene consistent occlusions may be formed

Occlusion in the original image means some parts of the model decomposition are unknown. We refer to these as *holes*. Holes constrain scene consistent retargeting: all holes must be kept occluded, to prevent the need to inpaint.

We use the following notation throughout. The image intensity is  $I_{r,c}$  for pixels  $(r, c)$  in the image domain  $\mathcal{P}$ . An object map is defined over the same domain as  $O(r, c) = o$  at pixels belonging to object  $o > 0$ ; otherwise,  $O(r, c) = 0$ .

## 4 Towards Scene Consistent Seam Carving

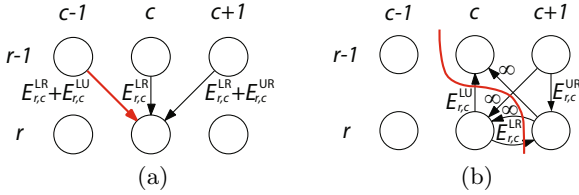
In this section, we recap seam carving (S.C.) (Sect. 4.1), which we extend to be able to create scene consistent object occlusions (Sect. 4.2) by enabling seams to pass through occlusion boundaries. This extension we call *seam carving with object occlusions* (S.C.+Obj. Occ.). We discuss a complication of this algorithm, namely that it does not easily allow for object reappearance, in Sect. 4.3.

### 4.1 Seam Carving

Our algorithms build on seam carving with forward energy [14]. Seam carving greedily removes seams with minimum energy from an image. A seam is an 8-connected path through the image, containing a single pixel on each row (assuming vertical seams are removed as we do throughout without loss of generality). Removing pixel  $(r, c)$  causes the following distortions: it brings into horizontal contact its **L**eft  $(r, c - 1)$  and **R**ight neighbours  $(r, c + 1)$  in row  $r$ . Depending on where the seam passed in row  $r - 1$ , it may additionally bring into vertical contact its **U**pper and **L**eft or its **U**pper and **R**ight neighbours. The energy of these distortions is captured in the following terms, used as illustrated in Fig. 3(a).

$$\begin{aligned}
 E_{r,c}^{\text{LR}} &= |I_{r,c-1} - I_{r,c+1}| \\
 E_{r,c}^{\text{LU}} &= |I_{r,c-1} - I_{r-1,c}| \\
 E_{r,c}^{\text{UR}} &= |I_{r-1,c} - I_{r,c+1}| .
 \end{aligned} \tag{1}$$

These terms measure distortion by magnitude similarity of neighbouring pixels.



**Fig. 3.** Graphs for dynamic programming (a) and graph cut (b) optimization of forward energy seam carving. Only terms related to pixel  $(r, c)$  are shown. The red arc in (a) corresponds to the red cut in (b), removing pixels  $(r - 1, c - 1)$  and  $(r, c)$

The seam that corresponds to minimal energy can be efficiently found using dynamic programming (D.P.), or using a graph cut (G.C.) [14]. In the latter, the problem is cast as a binary graph labelling problem. The corresponding graph is shown in Fig. 3(b). After the graph is cut, the pixel on each row directly left of the cut is the seam pixel, as exemplified by the red arc and cut in Fig. 3. These two frameworks are equivalent but have different properties [14].

The aim of this paper is to maintain scene consistency in retargeting. A simple method for preventing object distortion is given in [1], which we refer to as *seam carving with object protection* (S.C.+Obj. Prot.). The energies of all arcs pointing to pixels that belong to an object are set to infinity:

$$E_{r,c}^{\text{LR}} = E_{r,c}^{\text{LU}} = E_{r,c}^{\text{UR}} = \infty \quad \forall (r, c) \in \{(r, c) : O_{r,c} > 0\} . \quad (2)$$

This ensures that no seams pass through objects. As seams are progressively removed, objects are moved together until they abut. Continuing to remove seams, with infinite energy, would lead to great distortion (see Fig. 2(d)). For object consistency we enforce that seams may then pass only through edges of the image, resulting instead in a cropping (see Fig. 2(e)).

Neither of these methods allows seams to cut through the occlusion boundaries, moving objects behind one another. This would allow more flexibility for seams to be removed. In the next section we present an algorithm to do this.

## 4.2 Seam Carving with Object Occlusions

We now describe *seam carving with object occlusions* (S.C.+Obj. Occ.). This algorithm behaves like seam carving in background regions, but protects objects and allows seams to pass through occlusion boundaries between objects, as illustrated in Fig. 4(a). Two modifications are made, to the energies at occlusion boundaries and to the graph structure, with the use of “supernodes”.

**Occlusion Boundaries.** For background pixels that border the edge of the image or an object, the standard forward energy does not apply. Removing these pixels can be viewed as an occlusion, with no visual distortion created. We replace the energies for these pixels with a small value  $u_s = 10$ . For  $E_{r,c}^{\text{LR}}$ ,



$$E_{r,c}^{\text{LR}} = u_s \quad \forall (r, c) \in \{(r, c) : O_{r,c} = 0 \wedge ((r, c - 1) \notin \mathcal{P} \vee (r, c + 1) \notin \mathcal{P})\} \\ \cup \{(r, c) : O_{r,c} = 0 \wedge (O_{r,c-1} > 0 \vee O_{r,c+1} > 0)\} \quad (3)$$

gives the formal condition for use of this term, with similar definitions for  $E_{r,c}^{\text{LU}}$  and  $E_{r,c}^{\text{UR}}$ . This energy modification could also be applied to S.C.+Obj. Prot.

**Introducing Supernodes.** We must allow seams to run along object occlusion boundaries while protecting objects. With occlusions possible, object protection cannot be ensured by infinite energy terms as in (2). Consider the object in Fig. 4(a) that is occluded and separated into two parts. Consistency requires that seams pass all visible parts of an object on the same side, so seam (b) in the figure is invalid. As can be seen, consistency does not exhibit optimal substructure and cannot be optimized with dynamic programming.

We resolve this problem by considering the graph cut formulation and modifying the graph *structure* to protect objects. We introduce *supernodes*, nodes that subsume a group of pixel nodes. A supernode takes only a single label, so pixels subsumed by the supernode are assigned the same label.

Supernodes are constructed as follows, as illustrated in Fig. 4(b). Recall that in the graph cut formulation, the seam pixels are those directly left of the cut (c.f. Fig. 3(b)). We take the object closest to the camera and create a supernode from all object pixels as well as their right neighbours. This procedure is now iterated from the closest to the furthest object. At each step all object pixels and their right neighbours are included in the supernode, if they are not already in an existing supernode (e.g. the node in the second row, fourth column in 4(b)).

**Energy Terms for Supernodes.** The energy of object-background occlusion was defined in (3). We now define the energy of object-object occlusion. We set the energy terms of pixels in the occlusion boundary to a term  $u_o$  where

$$u_o = \frac{u_{\text{obj}}}{|\{(r, c) : O_{r,c} = o\}|} \quad (4)$$

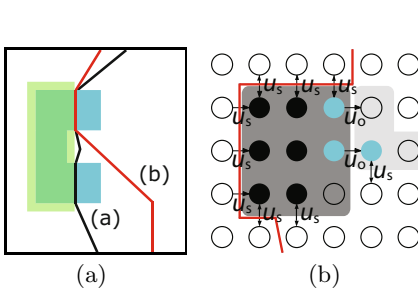
for a fixed constant  $u_{\text{obj}}$ . Setting this constant high increases the energy of occlusion of an object pixel, and even more so for smaller objects. We use  $u_{\text{obj}} = 10^7$ . Note that the borders of the image are treated in the same way, as an occluding object. Note also that if the occlusion is not valid, because it would lead to reappearance of part of an object behind another or because the objects next to each other are at the same depth, we can simply merge the supernodes for the two objects to prevent any further occlusion occurring.

Occlusion boundaries cannot be carved with the algorithm so far described if it is not possible for an 8-connected seam to pass through them. We therefore relax the connectivity constraint around objects, allowing seams to jump through horizontal occlusion boundaries. We do this by not attaching to supernodes the infinite cost arcs that enforce this constraint (e.g. the arc from  $(r - 1, c + 1) \rightarrow (r, c)$  in Fig. 3(b)). An example of a seam this allows is the red cut in Fig. 4(b).

### 4.3 Limitation to Non-reappearance

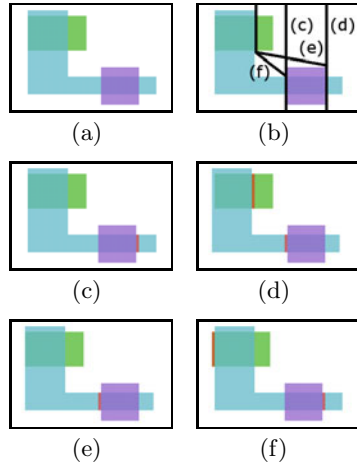
The described algorithm can only *remove* pixels, hence the need to prevent occlusions that would cause part of an object to reappear. We would like to relax this constraint and include an energy term for this reappearance, as in many images this is necessary to create useful occlusions as in Fig. 2(g). However, we found that extending the algorithm so far described to this would lead to an energy with higher order potentials, which are in general non-submodular and cannot be optimized efficiently. We demonstrate this with an example before describing, in Sect. 5, an algorithm which does not suffer this limitation.

Consider Fig. 5. Without reappearance, seams simply determine object movements: if the object is to the right of the seam, it is moved left, and if it is to the left, it maintains its position, relative to the left edge of the image. Without loss of generality we assume the same rule even with reappearance. The seams passing through the boundaries between the blue and purple objects simply determine the behaviour at this boundary: seams on the left (c) and (f) lead to reappearance on the right, and similarly with seams (d) and (e). Hence the reappearance energy can be associated with passing through the boundaries. However, no such relationship exists for the occlusion boundary between the blue and green objects (seams (e) and (f)), where the reappearance also depends on the purple object. In general, it would be necessary to encode the reappearance



**Fig. 4.** Left: The blue “C” shaped object is occluded (indicated by transparency) and thus split into two separate parts. Hence the red seam (b) does not preserve object consistency, while seam (a) does.

**Right:** Two objects, their corresponding supernodes and changed energy terms. The object with black pixels is closest and creates the supernode containing nodes in the dark grey area. The supernode of the blue object is the light shaded region. Also shown are those energy terms that changed compared to seam carving. The red line indicates a possible cut along the objects



**Fig. 5.** S.C.+Obj. Occ. with reappearance requires higher order terms. The seams passing through occlusion boundaries in (a) are shown in (b) resulting in (c) to (f). The objects are shown with transparency, with purple in front of blue in front of green. Reappearing pixels are highlighted in red

energy to depend on the positioning of all of the objects. This energy would contain higher-order potentials and in general be non-submodular.

## 5 Scene Carving

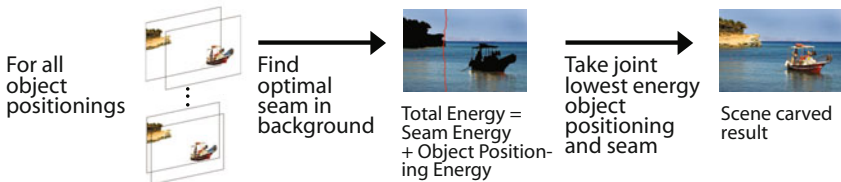
In seam carving, including the described extensions, the seam determines the movement of objects. This led to the problem that objects and background reappearance could not be optimized for efficiently. We resolve this problem by using a layered decomposition (Sect. 5.1) and adding the possibility of removing background holes (Sect. 5.2). This yields the *scene carving* (Sc. Carve) algorithm.

### 5.1 Layered Decomposition

The main idea of scene carving is the use of a layered image decomposition as illustrated in Fig. 1(c). Each object is stored in a separate layer. The last layer is referred to as the *background image*. This contains the background, with holes where the background is occluded by objects. From this representation an image can be created by “flattening” the layers onto the background image. Scene consistency is inherent if object layers are only translated in the plane, but have no pixels removed. We then only find seams in the background image.

This decomposition allows us to store an over-complete representation of the image. Pixels that are occluded in the flattened image are still stored in their respective layer and thus may reappear at a later iteration, as in Fig. 2(g).

The algorithm proceeds as shown in Fig. 6. At each iteration we consider all object positionings, and for each find the seam in the background image. Since the background image contains no objects, as in S.C., this can be done efficiently using dynamic programming. We calculate the total energy as the sum of the seam energy and object positioning energy, and take the joint minimum. Note that for  $V$  object movements and  $N$  objects, there are  $V^N$  object positionings to test at each iteration. We use the  $V = 2$  movements of S.C.: the object stays in the same position or moves one pixel to the left, relative to the left of the image. In Sect. 5.3 we describe a speed up for this combinatorial problem.



**Fig. 6.** Scene carving jointly optimizes for a new object positioning and a seam to be removed from the background image

## 5.2 Seams in the Background Image

Since the seam does not carry the burden of determining object movement it may pass anywhere in the background image, including through holes. The only restriction is to ensure that all holes are occluded in the resulting image. We now define the energy of such a seam in the background image.

**Distortion Domain.** We can distinguish two choices for seam energies, calculating the distortion of either (1) the flattened image or (2) the background only. S.C. advocates (1) and our extension in Sect. 4.2 also follows this rationale.

However strategy (1) comes at the expense of allowing high distortion to be created in the background image at no cost behind objects. This could severely limit our ability to move objects in further iterations and allow increased distortion in the background. Empirical results show that this occurs in our images. See Fig. 8(f) and 8(g), where this method is referred to as Sc. Carve-D<sup>2</sup>

We therefore take the second approach (2) and optimize at each iteration jointly for the highest fraction of objects to be visible and for the minimally distorting seam in the background image. This leads us to the *scene carving* algorithm. We pay the cost for introducing distortions that are not currently visible (but may be at future iterations), therefore sacrificing some potential improvement in the image at this iteration for a potentially better result image.

**Seam Energy.** We noted we find a seam only in the background image, so we are able to use D.P. for better runtime behaviour than S.C.+Obj. Occ. We construct the seam energy as follows. We reuse the graph of S.C. with the energies of (1). Energy terms for pixels next to the image boundary or holes are set as in (3) to a small constant, here  $u_s = 6$ . As seams may pass through holes, we set energy terms for hole pixels to a non-infinite constant  $u_h$ . Given a binary hole mask  $H$  taking value 0 where the background is known and 1 otherwise:

$$E_{r,c}^{\text{LR}} = E_{r,c}^{\text{LU}} = E_{r,c}^{\text{UR}} = u_h \quad \forall (r,c) \in \{(r,c) : H_{r,c} = 1\} . \quad (5)$$

We set  $u_h = 0$  to encourage removal of hole pixels.

Remaining hole pixels constrain object movement, as all must be kept occluded by an object. This constraint is ensured by setting the following energy:

$$E_{r,c}^{\text{LR}} = E_{r,c}^{\text{LU}} = E_{r,c}^{\text{UR}} = \infty \quad \forall r,c \in \{c : c > c_r^{\text{max}} \vee c < c_r^{\text{min}}\} \quad (6)$$

where:

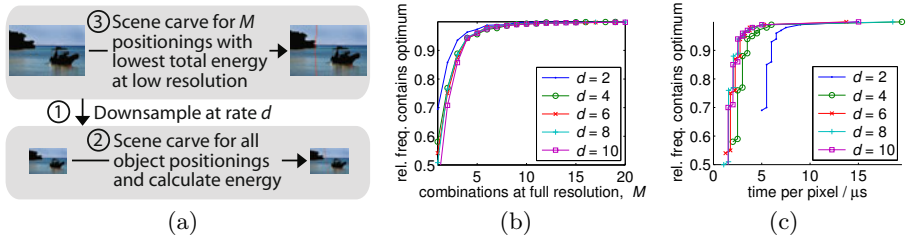
$$c_r^{\text{min}} = \max\{c : H_{r,c} = 1 \wedge O_{r,c} > 0 \wedge O_{r,c-1} = 0\}$$

$$c_r^{\text{max}} = \min\{c : H_{r,c} = 1 \wedge O_{r,c} = 0\} .$$

This constrains the seam at a row  $r$  to pass between the columns  $c_r^{\text{min}}$  and  $c_r^{\text{max}}$  <sup>3</sup>

<sup>2</sup> Details on how to define the energy for (1) and optimize it, taking all changes into account, can be found in the supplementary material, along with additional results.

<sup>3</sup> Small scale non-convexities in object segmentations can limit seams through this constraint, so we remove these by simple dilation and erosion processes.



**Fig. 7.** (a) Describes a 3 step hierarchical approximation to speed up scene carving. We set the parameters based on (b) and (c). (b) shows the relative frequency that the  $M$  object positionings checked in step ③ includes the optimal full resolution positioning, (c) the relative frequency against optimization time per pixel, assuming  $N = 5$  objects

**Object Positioning Energy.** We compute the *final energy* by adding to the energy of the optimal seam an object positioning energy term: the negative of the unary used in S.C.+Obj. Occ. (4). At each iteration we take the joint object positioning and background image seam with the lowest energy.

### 5.3 Speeding Up

Scene carving has computational complexity  $D.P. \times 2^N$  at each iteration. While dynamic programming is very efficient, this algorithm is still infeasible for large numbers of objects. We use two approaches to give a speed up.

Firstly, for a constant factor speed up, we note that objects only affect the energy on the rows they span, c.f. (6). We iterate through object positionings in a unit distance code, reusing the graph above and below the object moved.

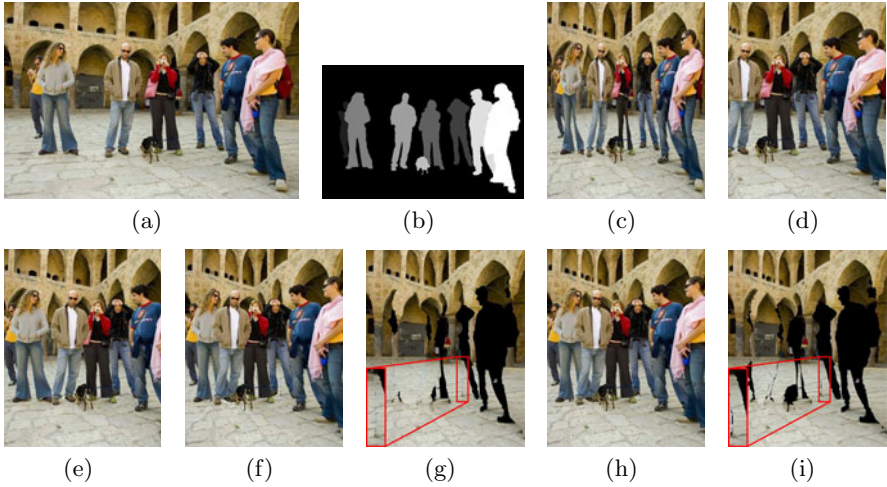
Secondly, we use a hierarchical speedup, as described in Fig. 7(a). We set  $M = 5$  and  $d = 6$  based on the following analysis. On 11 images containing 2-8 objects we removed 300 seams using scene carving at the full resolution and at lower resolutions. Our results are shown in Fig. 7(b) and Fig. 7(c). Choosing  $d = 6$  (red curve) and  $M = 5$  places us at the “knee” of the trade-off curves of Fig. 7(b). Here, the optimal object positioning is obtained approximately 97% of the time. Fig. 7(c) then shows that if we want to find the optimum approximately 97% of the time, greater downsampling would not increase the speed.

This method is still combinatorial in the number of objects, but with a lower multiplying factor. In most cases we expect a low number of objects to be labelled (up to 10), such that optimizing over all combinations of positionings is feasible.

## 6 Results

We now present results for our algorithms, and compare these results to those gained from our implementation of seam carving [4]. For convenience the key properties of these algorithms are summarized in Table 1.

<sup>4</sup> All code is available at [www.vision.ee.ethz.ch/~mansflea/scenecarving/](http://www.vision.ee.ethz.ch/~mansflea/scenecarving/) under the GNU General Public License.



**Fig. 8.** *People* image (a) with depth map (b) retargeted by S.C. (c), S.C.+Obj. Prot. (d), S.C.+Obj. Occ. (e), Sc. Carve-D. (f) with bkg. image (g), Sc. Carve (h) with bkg. image (i) (300 seams removed). Note the distortion introduced by S.C., and cropping with S.C.+Obj. Prot. and S.C.+Obj. Occ. Sc. Carve keeps all objects, with the red boxes highlighting background distortion from Sc. Carve-D. not in Sc. Carve

The power of our algorithms can be demonstrated with the example of the *People* image (from [11]) in Fig. 8. Seam carving (Fig. 8(c)) can be seen to create visually disturbing distortion of the people. Ensuring object consistency prevents this, but because there is no occlusion handling, this results in a cropped image with the two left-most people removed completely. (Fig. 8(d)).

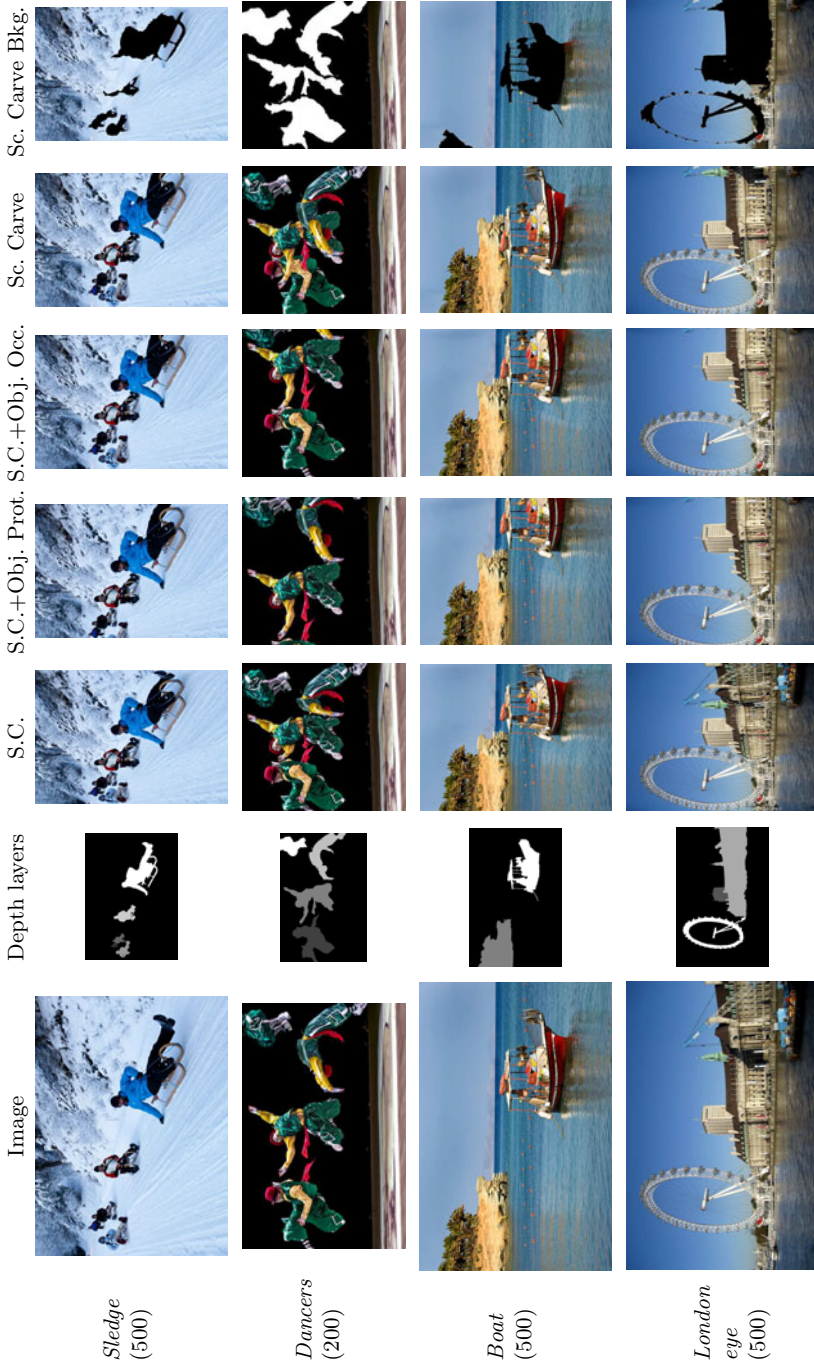
Our algorithms guarantee scene consistency. S.C.+Obj. Occ. (Fig. 8(e)) moves the people together until reappearance would occur. Further seams are removed at the edges of the image, again cropping one person out. With Sc. Carve-D., reappearance is possible, but the ability to hide high gradients behind parts of objects allows distortion to be created in the background image, which are visible in the resulting image. These distortions are shown in Fig. 8(g), highlighted in the red box. Scene carving (Fig. 8(h)) is able to keep all people in the image, scene consistently, combined with a pleasing background (Fig. 8(i)).

Further results, demonstrating the same effects, are shown in Fig. 9.

Limitations of our methods can also be seen in these images. For example, in the *Boat* image, it can be seen that our freedom to edit the background image

**Table 1.** Properties of the algorithms tested

	Scene consistent	Creates occlusions	With re-appearance	Optimization
S.C. [1, 14]	×	×	×	DP or GC
S.C.+Obj. Prot. [1, 14]	✓	×	×	DP or GC
S.C.+Obj. Occ. (Sect. 4.2)	✓	✓	×	GC
Sc. Carve (Sect. 5)	✓	✓	✓	DP( $5 + 2^N/6^2$ )



**Fig. 9.** Further results. The number by each image name indicates the number of seams removed. The final column of images shows the background image at the end of Sc. Carve. Holes are shown in black for all images, except for *Dancers* where they are shown in white

**Table 2.** Time taken to produce results with our *Matlab/Mex* implementation

	<i>People</i>	<i>Sledge</i>	<i>Dancers</i>	<i>London eye</i>	<i>Boat</i>
No. objects	8	5	4	3	2
Size	640 × 427	1024 × 759	500 × 333	1024 × 683	1016 × 677
No. seams removed	300	500	200	500	500
S.C. [1, 14]	22s	62s	14s	49s	64s
S.C.+Obj. Prot. [1, 14]	28s	69s	9s	60s	54s
S.C.+Obj. Occ. (Sect. 4.2)	4515s	46152s	328s	2079s	19941s
Sc. Carve (Sect. 5)	352s	596s	73s	711s	619s

has shrunk the boat reflection so it no longer spans the whole boat. Another effect, caused by inaccurate segmentations, is shown in the *London eye* image, where sky can be seen through the wheel, where the building should be visible.

Table 2 shows the time taken to produce our results. In all cases, Sc. Carve is the fastest algorithm that allows for object occlusions. S.C.+Obj. Occ., while a non-combinatorial optimization problem, in practice produces a graph that is slow to optimize. S.C. and S.C.+Obj. Prot. are much faster, but may respectively lead to object distortion, or cropping and bad background distortion.

## 7 Conclusions and Future Work

In this work we considered the problem of scene consistent image retargeting. We developed two algorithms to perform such image retargeting, given a relative depth map: *seam carving with object occlusions* and *scene carving*.

The former was derived by making use of supernodes, enabling correct occlusion handling for the first time. This algorithm has the appealing property of requiring a single optimization in each step. However, accounting for reappearing material leads to graphs which cannot be optimized efficiently. Even without reappearance, the graph can be slow to optimize in practice.

Scene carving utilizes a layered decomposition of the image to allow flexible object re-arrangement. We find the joint global optimum seam and re-arrangement at each iteration with dynamic programming, at the expense of an overall combinatorial problem. We presented a more efficient hierarchical approximation, which still finds the global optimal in almost all iterations.

In summary, we recommend scene carving as the better algorithm, given that it is usually faster and produces visually superior results. Seam carving with occlusions may be competitive only when very many objects are present.

There are several possible routes to be followed. First we want to automate relative depth map creation using either high-level computer vision such as object detection, or stereo vision. Also, the seam carving algorithm can be understood as “forgetting” the previous input at each iteration. Other methods optimize an energy defined between the input and output image [5, 11, 15, 18, 19, 22]. We plan to derive a similar retargeting method for our problem scenario.

**Acknowledgements.** We thank the following users of Flickr for placing their work under the Creative Commons License: badkleinkirchheim (*Sledge*), William Hamon (*Dancers*), Alain Bachellier (*Boat*), wallyg (*London eye*).



## References

1. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. In: SIGGRAPH (2007)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. In: SIGGRAPH (2009)
3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23(11) (2001)
4. Cho, T.S., Butman, M., Avidan, S., Freeman, W.: The patch transform and its applications to image editing. In: CVPR (2008)
5. Dong, W., Zhou, N., Paul, J.C., Zhang, X.: Optimized image resizing using seam carving and scaling. ACM Trans. Graph. 28(5) (2009)
6. Gal, R., Sorkine, O., Cohen-Or, D.: Feature-aware texturing. In: Eurographics Symposium on Rendering (2006)
7. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Discontinuous seam-carving for video retargeting. In: CVPR (2010)
8. Han, D., Wu, X., Sonka, M.: Optimal multiple surfaces searching for video/image resizing - a graph-theoretic approach. In: ICCV (2009)
9. Hoiem, D., Stein, A., Efros, A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV (2007)
10. Krähenbühl, P., Lang, M., Hornung, A., Gross, M.: A system for retargeting of streaming video. In: SIGGRAPH (2009)
11. Pritch, Y., Kav-Venaki, E., Peleg, S.: Shift-map image editing. In: ICCV (2009)
12. Rav-Acha, A., Pritch, Y., Shmuel, P.: Making a long video short: Dynamic video synopsis. In: CVPR (2006)
13. Rother, C., Kolmogorov, V., Blake, A.: “GrabCut”: interactive foreground extraction using iterated graph cuts. In: SIGGRAPH (2004)
14. Rubinstein, M., Shamir, A., Avidan, S.: Improved seam carving for video retargeting. In: SIGGRAPH (2008)
15. Rubinstein, M., Shamir, A., Avidan, S.: Multi-operator media retargeting. ACM Trans. Graph. 28(3) (2009)
16. Saxena, A., Sun, M., Ng, A.: Make3d: Learning 3-d scene structure from a single still image. IEEE PAMI 31(5) (2009)
17. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vision 47(1-3) (2002)
18. Setlur, V., Takagi, S., Raskar, R., Gleicher, M., Gooch, B.: Automatic image retargeting. In: Int. Conf. on Mobile and Ubiquitous Multimedia (2005)
19. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: CVPR (2008)
20. Suh, B., Ling, H., Bederson, B.B., Jacobs, D.W.: Automatic thumbnail cropping and its effectiveness. In: UIT: ACM Symposium on User Interface Software and Technology (2003)
21. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. PAMI 30(6) (2008)
22. Wang, Y.S., Tai, C.L., Sorkine, O., Lee, T.Y.: Optimized scale-and-stretch for image resizing. In: SIGGRAPH Asia (2008)
23. Wolf, L., Guttman, M., Cohen-Or, D.: Non-homogeneous content-driven video-retargeting. In: ICCV (2007)

# Two-Phase Kernel Estimation for Robust Motion Deblurring

Li Xu and Jiaya Jia

Department of Computer Science and Engineering  
The Chinese University of Hong Kong  
{xuli,leo[jia]}@cse.cuhk.edu.hk

**Abstract.** We discuss a few new motion deblurring problems that are significant to kernel estimation and non-blind deconvolution. We found that strong edges do not always profit kernel estimation, but instead under certain circumstance degrade it. This finding leads to a new metric to measure the usefulness of image edges in motion deblurring and a gradient selection process to mitigate their possible adverse effect. We also propose an efficient and high-quality kernel estimation method based on using the spatial prior and the iterative support detection (ISD) kernel refinement, which avoids hard threshold of the kernel elements to enforce sparsity. We employ the TV- $\ell_1$  deconvolution model, solved with a new variable substitution scheme to robustly suppress noise.

## 1 Introduction

Motion deblurring was hotly discussed in the computer vision and graphics community due to its involvement of many challenges in problem formulation, regularization, and optimization. Notable progress has been made lately [1–6]. The blur process caused by camera shake is generally modeled as a latent image convolved with a blur point-spread-function (a.k.a. kernel).

The success of recent single-image methods partly stems from the use of various sparse priors, for either the latent images or motion blur kernels [1, 3, 6]. It was found that without these constraints, iterative kernel estimation is easily stuck in local minima and possibly results in a dense kernel and many visual artifacts in the restored image. However, minimizing a non-convex energy function with the kernel-sparsity prior is usually costly.

Another group of methods seek high efficiency and resort to explicitly detecting salient image structures. They use the Gaussian kernel priors [4, 5, 7] instead of the sparse ones. These approaches greatly shorten the computation time; but the Gaussian priors sometimes issue in noisy or dense kernel estimates, which need to be post-processed by threshold-like operations.

Despite the efficiency and accuracy issues, another critical motion deblurring problem that was not known yet is on how image structure influences kernel estimation. Our intriguing finding is that salient edges do not always help kernel refinement, but instead in some commonly encountered circumstances greatly

increase the estimation ambiguity. We will analyze this problem and propose an automatic gradient selection algorithm to exclude the detrimental structures.

Our method also makes several other contributions. 1) First, we propose a novel two-phase kernel estimation algorithm to separate computationally expensive non-convex optimization from quick kernel initialization, giving rise to an efficient and robust kernel estimation process. 2) We introduce a new spatial prior to preserve sharp edges in quick latent image restoration. 3) In the kernel refinement stage, we employ the Iterative Support Detection (ISD) algorithm, which is a powerful numerical scheme through iterative support detection, to adaptively enforce the sparsity constraint and properly preserve large-value elements. Soft-threshold-like effect is achieved in this step. 4) Finally, to restore the latent image, we employ a TV- $\ell_1$  objective function that is robust to noise and develop an efficient solver based on half-quadratic splitting.

We applied our method to challenging examples, where many images are blurred with very large PSFs (spanning up to 100 pixels in width or height) due to camera shake. Our “robust deblurring” project website is put online<sup>1</sup>, which includes the motion deblurring executable and image data.

## 1.1 Related Work

Shift-invariant motion blur can be modeled as image convolution with a PSF. We briefly review the blind and non-blind deconvolution methods.

*Blind Deconvolution.* Early work on blind image deconvolution focuses on estimating small-size blur kernels. For example, You and Kaveh [8] proposed a variational framework to estimate small Gaussian kernels. Chan and Wong [9] applied the Total Variation regularizers to both kernels and images. Another group of methods [10–12] did not compute the blur kernels, but studied the reversion of a diffusion process.

Lately, impressive progress has been made in estimating a complex motion blur PSF from a single image [1, 3, 6]. The success arises in part from the employment of sparse priors and the multi-scale framework. Fergus *et al.* [1] used a zero-mean Mixture of Gaussian to fit the heavy-tailed natural image prior. A variational Bayesian framework was employed. Shan *et al.* [3] also exploited the sparse priors for both the latent image and blur kernel. Deblurring is achieved through an alternating-minimization scheme. Cai *et al.* [6] introduced a framelet and curvelet system to obtain the sparse representation for kernels and images. Levin *et al.* [13] showed that common MAP methods involving estimating both the image and kernel likely fail because they favor the trivial solution. Special attention such as edge re-weighting is probably the remedy. It is notable that using sparse priors usually result in non-convex objective functions, encumbering efficient optimization.

Another group of methods [4, 5, 7] do not use sparse priors, but instead employ an explicit edge prediction step for the PSF estimation. Specifically, Joshi *et al.* [4] predicted sharp edges by first locating step edges and then propagating

<sup>1</sup> [http://www.cse.cuhk.edu.hk/~leojia/projects/robust\\_deblur/index.html](http://www.cse.cuhk.edu.hk/~leojia/projects/robust_deblur/index.html)

the local intensity extrema towards the edge. This method was used to handle complex PSFs with a multi-scale scheme [7]. Cho and Lee [5] adopted bilateral filtering together with shock filtering to predict sharp edges. These methods impose simple Gaussian priors, which avail to construct quick solvers. These priors however cannot capture the sparse nature of the PSF and image structures, which occasionally make the estimates noisy and dense.

*Non-blind deconvolution.* Given a known blur PSF, the process of restoring an unblurred image is referred to as non-blind deconvolution. Early work such as Richardson-Lucy (RL) or Weiner filtering is known as sensitive to noise. Yuan *et al.* [14] proposed a progressive multi-scale refinement scheme based on an edge-preserving bilateral Richardson-Lucy (BRL) method. Total Variation regularizer (also referred to as Laplacian prior) [9], heavy-tailed natural image priors [1, 3] and Hyper-Laplacian priors [15–18] were also extensively studied.

To suppress noise, Bar *et al.* [19] used the  $\ell_1$  fidelity term together with a Mumford-Shah regularizer to reject impulse noise. Joshi *et al.* [20] incorporated a local two-color prior to suppress noise. These methods used the iterative re-weighted least square to solve the nonlinear optimization problem, which inevitably involves intensive computation. In this paper, we developed a fast TV- $\ell_1$  deconvolution method based on half-quadratic splitting [16, 18], to efficiently reject outliers and preserve structures.

## 2 Two-Phase Sparse Kernel Estimation

By convention, the blur process is modeled as

$$B = I \otimes k + \varepsilon,$$

where  $I$  is the latent image,  $k$  is the blur kernel,  $\varepsilon$  is the image noise,  $\otimes$  denotes convolution and  $B$  is the observed blur image. In this section, we introduce a two-phase method for PSF estimation. The first stage aims to efficiently compute a coarse version of the kernel without enforcing much sparsity. In the second phase, although non-convex optimization is employed, with the initial kernel estimate propagated from stage one, no significant computation is required to produce the final result.

### 2.1 Phase One: Kernel Initialization

In the first step, we estimate the blur kernel in a multi-scale setting. High efficiency can be yielded as we use the Gaussian priors where closed-form solutions exist. The algorithm is sketched in Alg. 1 with three main steps – that is, sharp edge construction, kernel estimation, and coarse image restoration.

In the first place, like other motion deblurring methods, we filter the image and predict salient edges to guide the kernel initialization. We use Gaussian

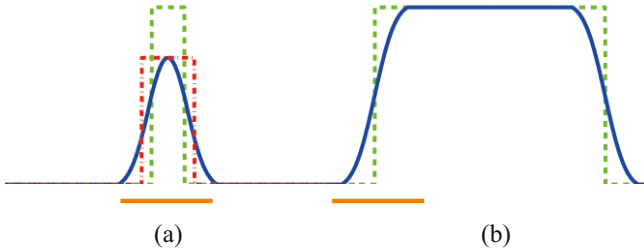
---

**Algorithm 1.** Kernel Initialization

---

**INPUT:** Blur image B and an all-zero kernel (size  $h \times h$ )  
 Build an image pyramid with level index  $\{1, 2, \dots, n\}$ .  
**for**  $l = 1$  to  $n$  **do**  
     Compute gradient confidence  $r$  for all pixels (Eq. (2)).  
     **for**  $i = 1$  to  $m$  ( $m$  is the number of iterations) **do**  
         (a) Select edges  $\nabla I^s$  for kernel estimation based on confidence  $r$  (Eq. (4)).  
         (b) Estimate kernel with the Gaussian prior (Eq. (6)).  
         (c) Estimate the latent image  $I^l$  with the spatial prior (Eq. (8)), and update  
              $\tau_s \leftarrow \tau_s/1.1, \tau_r \leftarrow \tau_r/1.1$ .  
     **end for**  
     Upscale image  $I^{l+1} \leftarrow I^l \uparrow$ .  
**end for**  
**OUTPUT:** Kernel estimate  $k^0$  and sharp edge gradient map  $\nabla I^s$

---



**Fig. 1.** Ambiguity in motion deblurring. Two latent signals (green dashed lines) in (a) and (b) are blurred (shown in blue) with the same Gaussian kernel. In (a), the blurred signal is not total-variation preserving, making the kernel estimation ambiguous. In fact, the red curve is more likely the latent signal than the green one in a common optimization process. The bottom orange lines indicate the input kernel width.

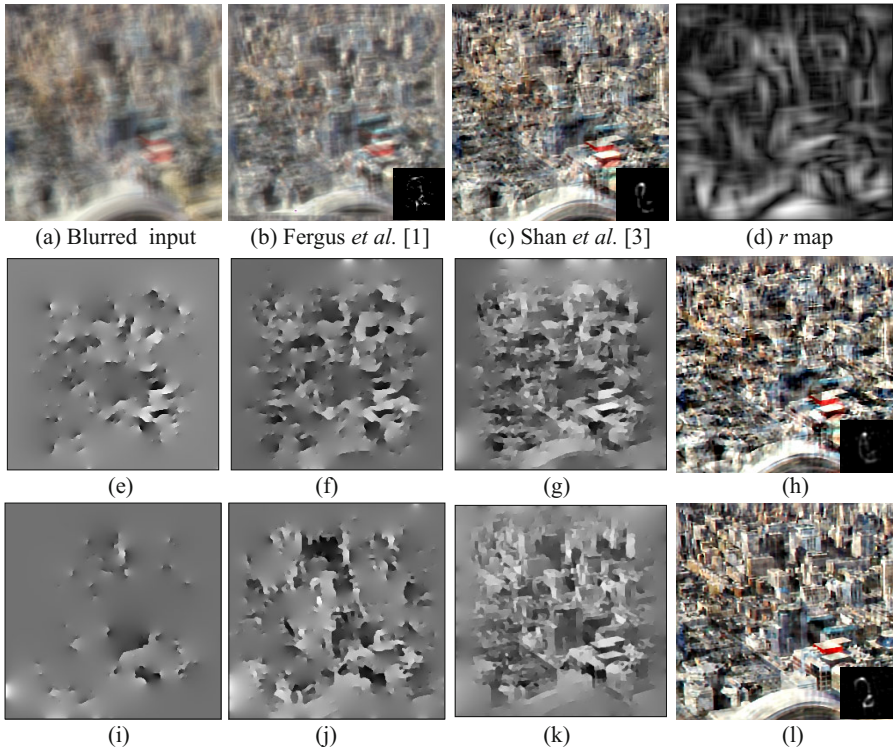
filtering to pre-smooth the image and then solve the following shock filtering PDE problem [10] to construct significant step edges:

$$\partial I / \partial t = -\text{sign}(\Delta I) \|\nabla I\|, \quad I_0 = G_\sigma \otimes I_{input}, \quad (1)$$

where  $\nabla I = (I_x, I_y)'$  and  $\Delta I = I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}$  are the first- and second-order spatial derivatives respectively.  $I_0$  denotes the Gaussian smoothed input image, which serves as an initial input for iteratively updating  $\partial I / \partial t$ .

**Selective Edge Map for Kernel Estimation.** Insignificant edges make PSF estimation vulnerable to noise, as discussed in [3–5, 13]. We however observe a different connection between image edges and the quality of kernel estimation – that is, salient edges do not always improve kernel estimation; on the contrary, *if the scale of an object is smaller than that of the blur kernel, the edge information could damage kernel estimation.*

We give an example in Figure 1. Two step signals (the green dashed lines) in (a) and (b) are blurred with a large PSF. The observed blur signals are shown



**Fig. 2.** Image structure influence in kernel estimation. (a) Blurred image. (b) Result of Fergus *et al.* [1]. (c) Result of Shan *et al.* [3]. (d)  $r$  map (by Eq. (2)). (e)-(g)  $\nabla I^s$  maps, visualized using Poisson reconstruction, in the 1st, 2nd and 7th iterations without considering  $r$ . (h) Deblurring result not using the  $r$  map. (i)-(k)  $\nabla I^s$  maps computed according to Eq. (4). (l) Our final result. The blur PSF is of size  $45 \times 45$ .

in blue. Because the left signal is horizontally narrow, the blur process lowers its height in (a), yielding ambiguity in the latent signal restoration. Specifically, motion blur methods imposing sparse priors on the gradient map of the latent image [1, 3] will favor the red dashed line in computing the unblurred signal because this version presents smaller gradient magnitudes. Moreover, the red signal preserves the total variation better than the green one. So it is also a more appropriate solution for the group of methods using sharp edge prediction (including shock filtering and the method of [4]). This example shows that if image structure magnitude significantly changes after blur, the corresponding edge information could mistake kernel estimation.

In comparison, the larger-scale object shown in Figure 1(b) can yield stable kernel estimation because it is wider than the kernel, preserving the total variation of the latent signal along its edges.

Figure 2 shows an image example. The blurred input (shown in (a)) contains rich edge information along many small-scale objects. The results of Fergus *et*

al. [1] (b) and Shan *et al.* [3] (c) are computed by extensively hand-tuning parameters. However, the correct kernel estimate still cannot be found, primarily due to the aforementioned small structure problem.

We propose a new criterion for selecting informative edges for kernel estimation. The new metric to measure the usefulness of gradients is defined as

$$r(x) = \frac{\|\sum_{y \in N_h(x)} \nabla B(y)\|}{\sum_{y \in N_h(x)} \|\nabla B(y)\| + 0.5}, \quad (2)$$

where  $B$  denotes the blurred image and  $N_h(x)$  is a  $h \times h$  window centered at pixel  $x$ . 0.5 is to prevent producing a large  $r$  in flat regions. The signed  $\nabla B(y)$  for narrow objects (spikes) will mostly cancel out in  $\|\sum_{y \in N_h(x)} \nabla B(y)\|$ .  $\sum_{y \in N_h(x)} \|\nabla B(y)\|$  is the sum of the absolute gradient magnitudes in  $N_h(x)$ , which estimates how strong the image structure is in the window. A small  $r$  implies that either spikes or a flat region is involved, which causes neutralizing many gradient components. Figure 2(d) shows the computed  $r$  map.

We then rule out pixels belonging to small  $r$ -value windows using a mask

$$M = H(r - \tau_r), \quad (3)$$

where  $H(\cdot)$  is the Heaviside step function, outputting zeros for negative values and ones otherwise.  $\tau_r$  is a threshold. The final selected edges for kernel estimation are determined as

$$\nabla I^s = \tilde{\nabla I} \cdot H(M \|\tilde{\nabla I}\|_2 - \tau_s), \quad (4)$$

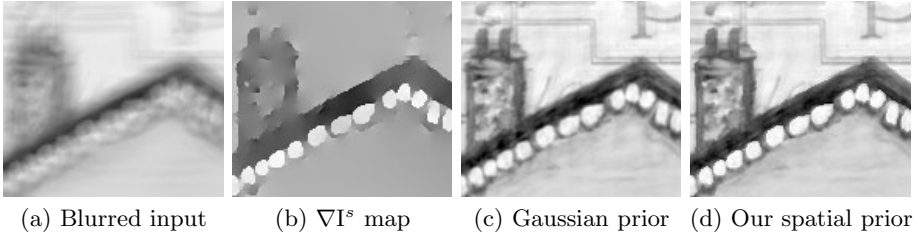
where  $\tilde{\nabla I}$  denotes the shock filtered image and  $\tau_s$  is a threshold of the gradient magnitude. Eq. (4) excludes part of the gradients, depending jointly on the magnitude  $\|\tilde{\nabla I}\|_2$  and the prior mask  $M$ . This selection process reduces ambiguity in the following kernel estimation.

Figures 2(e)-(g) and (i)-(k) illustrate the computed  $\nabla I^s$  maps in different iterations without and with the edge selection operation. The comparison shows that including more edges do not necessarily benefit kernel estimation. Optimization could be misled especially in the first a few iterations. So an image edge selection process is vital to reduce the confusion.

To allow for inferring subtle structures during kernel refinement, we decrease the values of  $\tau_r$  and  $\tau_s$  in iterations (divided by 1.1 in each pass), to include more and more edges. So the maps in (g) and (k) contain similar amount of edges. But the quality notably differs. The method to compute the final results shown in (h) and (l) is detailed further below.

**Fast Kernel Estimation.** With the critical edge selection, initial kernel estimation can be accomplished quickly. We define the objective function with a Gaussian regularizer as

$$E(k) = \|\nabla I^s \otimes k - \nabla B\|^2 + \gamma \|k\|^2, \quad (5)$$



**Fig. 3.** Comparison of results using the sparse  $\|\nabla\mathbf{I}\|^2$  and spatial  $\|\nabla\mathbf{I} - \nabla\mathbf{I}^s\|^2$  priors. The spatial prior makes the result in (d) preserve more sharp edges.

where  $\gamma$  is a weight. Based on the Parseval's theorem, we perform FFTs on all variables and set the derivative w.r.t.  $k$  to zero. The closed-form solution is given by

$$k = \mathcal{F}^{-1} \left( \frac{\overline{\mathcal{F}(\partial_x \mathbf{I}^s)} \mathcal{F}(\partial_x \mathbf{B}) + \overline{\mathcal{F}(\partial_y \mathbf{I}^s)} \mathcal{F}(\partial_y \mathbf{B})}{\mathcal{F}(\partial_x \mathbf{I}^s)^2 + \mathcal{F}(\partial_y \mathbf{I}^s)^2 + \gamma} \right), \quad (6)$$

where  $\mathcal{F}(\cdot)$  and  $\mathcal{F}^{-1}(\cdot)$  denote the FFT and inverse FFT respectively.  $\overline{\mathcal{F}(\cdot)}$  is the complex conjugate operator.

**Coarse Image Estimation with a Spatial Prior** We use the predicted sharp edge gradient  $\nabla\mathbf{I}^s$  as a spatial prior to guide the recovery of a coarse version of the latent image. The objective function is

$$E(\mathbf{I}) = \|\mathbf{I} \otimes k - \mathbf{B}\|^2 + \lambda \|\nabla\mathbf{I} - \nabla\mathbf{I}^s\|^2, \quad (7)$$

where the new spatial prior  $\|\nabla\mathbf{I} - \nabla\mathbf{I}^s\|^2$  does not blindly enforce small gradients near strong edges and thus allows for a sharp recovery even with the Gaussian regularizer. The closed-form solution exists. With a few algebraic operations in the frequency domain, we obtain

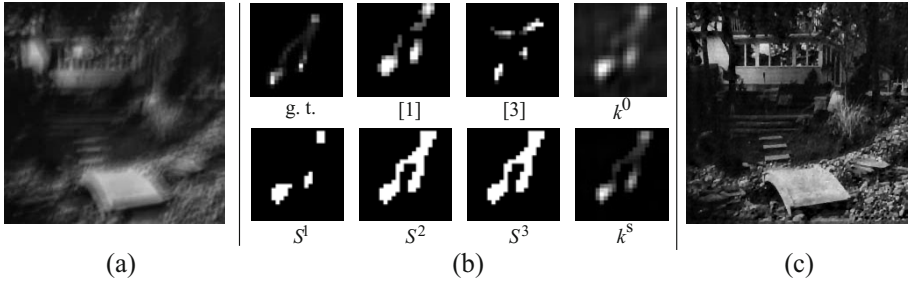
$$\mathbf{I} = \mathcal{F}^{-1} \left( \frac{\overline{\mathcal{F}(k)} \mathcal{F}(\mathbf{B}) + \lambda (\overline{\mathcal{F}(\partial_x)} \mathcal{F}(\mathbf{I}_x^s) + \overline{\mathcal{F}(\partial_y)} \mathcal{F}(\mathbf{I}_y^s))}{\overline{\mathcal{F}(k)} \mathcal{F}(k) + \lambda (\overline{\mathcal{F}(\partial_x)} \mathcal{F}(\partial_x) + \overline{\mathcal{F}(\partial_y)} \mathcal{F}(\partial_y))} \right). \quad (8)$$

Figure 3 compares the deconvolution results using the spatial and Gaussian priors respectively (the latter is usually written as  $\|\nabla\mathbf{I}\|^2$ ). The regularization weight  $\lambda = 2e^{-3}$ . The image shown in Figure 3(d) contains well preserved sharp edges.

## 2.2 Phase Two: ISD-Based Kernel Refinement

To obtain sparse PSFs, previous methods [1, 3, 5, 21] apply hard or hysteresis thresholding to the kernel estimates. These operations however ignore the inherent blur structure, possibly degrading the kernel quality. One example is shown





**Fig. 4.** Sparse Kernel Refinement. (a) A blurred image [13]. (b) Kernels. The top row shows respectively the ground truth kernel, the kernel estimates of Fergus *et al.* [1], Shan *et al.* [3], and of our method in phase one.  $k^s$  in the bottom row is our final result after kernel refinement.  $S^1$ - $S^3$  show the iteratively detected support regions by the ISD method. (c) Our restored image using  $k^s$ .

in Figure 4(b), where only keeping the large-value elements apparently cannot preserve the subtle structure of the motion PSF.

We solve this problem using an iterative support detection (ISD) method that can ensure the deblurring quality while removing noise. The idea is to iteratively secure the PSF elements with large values by relaxing the regularization penalty. So these elements will not be significantly affected by regularization in the next-round kernel refinement. This strategy was shown in [22] capable of correcting imperfect estimates and converging quickly.

ISD is an iterative method. At the beginning of each iteration, previously estimated kernel  $k^i$  is used to form a partial support; that is, large-value elements are put into a set  $S^{i+1}$  and all others belong to the set  $\overline{S^{i+1}}$ .  $S^{i+1}$  is constructed as

$$S^{i+1} \leftarrow \{j : k_j^i > \epsilon^s\}, \quad (9)$$

where  $j$  indexes the elements in  $k^i$  and  $\epsilon^s$  is a positive number, evolving in iterations, to form the partial support. We configure  $\epsilon^s$  by applying the “first significant jump” rule [22]. Briefly speaking, we sort all elements in  $k^i$  in an ascending order w.r.t. their values and compute the differences  $d_0, d_1 \dots$  between each two nearby elements. Then we exam these differences sequentially starting from the head  $d_0$  and search for the first element,  $d_j$  for example, that satisfies  $d_j > \|k^i\|_\infty / (2h \cdot i)$ , where  $h$  is the kernel width and  $\|k^i\|_\infty$  returns the largest value in  $k^i$ . We then assign the kernel value in position  $j$  to  $\epsilon^s$ . More details are presented in [22]. Examples of the detected support are shown in the bottom row of Figure 4(b). The elements within each  $S$  will be less penalized in optimization, resulting in an adaptive kernel refinement process.

We then minimize

$$E(k) = \frac{1}{2} \|\nabla I^s \otimes k - \nabla B\|^2 + \gamma \sum_{j \in S^{i+1}} |k_j| \quad (10)$$

**Algorithm 2.** ISD-based Kernel Refinement

---

**INPUT:** Initial kernel  $k^0$ ,  $\nabla B$ , and  $\nabla I^s$  (output of Algorithm 1)  
Initialize the partial support  $\bar{S}^0$  on  $k^0$  (Eq. (9)).  
**repeat**  
  Solve for  $k^i$  by minimizing Eq. (10).  
  Update  $\bar{S}$  (Eq. (9)).  
   $i \leftarrow i + 1$ .  
**until**  $\frac{\|k^{i+1} - k^i\|}{\|k^i\|} \leq \epsilon_k$  ( $\epsilon_k = 1e^{-3}$  empirically)  
**OUTPUT:** Kernel estimate  $k^s$

---

for PSF refinement. The difference between this function and those used in [3, 6] is on the definition of the regularization terms. Thresholding applies softly in our function through adaptive regularization, which allows the energy to concentrate on significant values and thus automatically maintains PSF sparsity, faithful to the deblurring process. The algorithm is outlined in Alg. 2.

To minimize Eq. (10) with the partial support, we employed the iterative reweighted least square (IRLS) method. By writing convolution as matrix multiplication, the latent image  $I$ , kernel  $k$ , and blur input  $B$  are correspondingly expressed as matrix  $A$ , vector  $V_k$ , and vector  $V_B$ . Eq. (10) is then minimized by iteratively solving linear equations w.r.t.  $V_k$ . In the  $t$ -th pass, the corresponding linear equation is expressed as

$$[A^T A + \gamma \text{diag}(V_{\bar{S}} \Psi^{-1})] V_k^t = A^T V_B, \quad (11)$$

where  $A^T$  denotes the transposed version of  $A$  and  $V_{\bar{S}}$  is the vector form of  $\bar{S}$ .  $\Psi$  is defined as  $\Psi = \max(\|V_k^{t-1}\|_1, 1e^{-5})$ , which is the weight related to the kernel estimate from the previous iteration.  $\text{diag}(\cdot)$  produces a diagonal matrix from the input vector. Eq. (11) can be solved by the conjugate gradient method in each pass (we alternatively apply the matrix division operation in Matlab). As PSFs have small size compared to images, the computation is very fast.

Our final kernel result  $k^s$  is shown in Figure 4(b). It maintains many small-value elements; meanwhile, the structure is appropriately sparse. Optimization in this phase converges in only a few iterations. Figure 4(c) shows our restored image using the computed PSF. It contains correctly reconstructed textures and small edges, verifying the quality of the kernel estimate.

### 3 Fast TV- $\ell_1$ Deconvolution

Assuming the data fitting costs following a Gaussian distribution is not a good way to go in many cases. It possibly makes results vulnerable to outliers, as demonstrated in many literatures. To achieve high robustness, we propose a TV- $\ell_1$  model in deconvolution, which is written as

$$E(I) = \|I \otimes k - B\| + \lambda \|\nabla I\|. \quad (12)$$

---

**Algorithm 3.** Robust Deconvolution

---

**INPUT:** Blurred image  $B$  and the estimated kernel  $k^s$

Edge taping in Matlab

$I \leftarrow B, \beta \leftarrow \beta_0.$

**repeat**

Solve for  $v$  using Eq. (18)

$\theta \leftarrow \theta_0$

**repeat**

Solve for  $w$  using Eq. (17)

Solve for  $I$  in the frequency domain using Eq. (15)

$\theta \leftarrow \theta/2$

**until**  $\theta < \theta_{\min}$

$\beta \leftarrow \beta/2$

**until**  $\beta < \beta_{\min}$

**OUTPUT:** Deblurred image  $I$

---

It contains non-linear penalties for both the data and regularization terms. We propose solving it using an efficient alternating minimization method, based on a half-quadratic splitting for  $\ell_1$  minimization [16, 18].

For each pixel, we introduce a variable  $v$  to equal the measure  $I \otimes k - B$ . We also denote by  $w = (w_x, w_y)$  image gradients in two directions. The use of these auxiliary variables leads to a modified objective function

$$E(I, w, v) = \frac{1}{2\beta} \|I \otimes k - B - v\|^2 + \frac{1}{2\theta} \|\nabla I - w\|_2^2 + \|v\| + \lambda \|w\|, \quad (13)$$

where the first two terms are used to ensure the similarity between the measures and the corresponding auxiliary variables. When  $\beta \rightarrow 0$  and  $\theta \rightarrow 0$ , the solution of Eq. (13) approaches that of Eq. (12).

With the adjusted formulation, Eq. (13) can now be solved by an efficient Alternating Minimization (AM) method, where the solver iterates among solving  $I$ ,  $w$ , and  $v$  independently by fixing other variables.  $w$  and  $v$  are initialized to zeros.

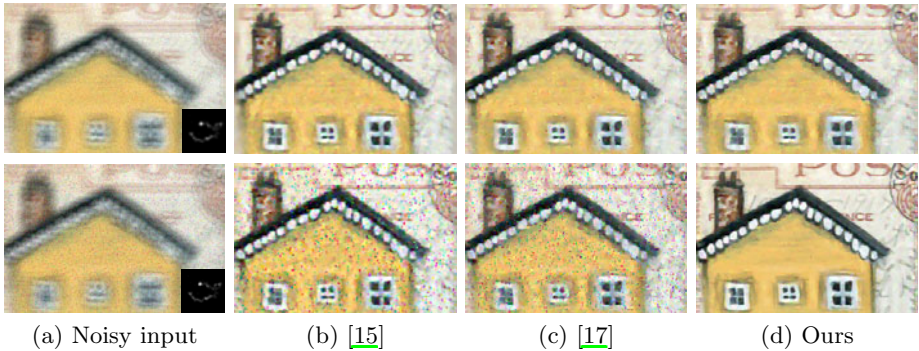
In each iteration, we first compute  $I$  given the initial or estimated  $w$  and  $v$  by minimizing

$$E(I; w, v) = \|I \otimes k - B - v\|^2 + \frac{\beta}{\theta} \|\nabla I - w\|_2^2. \quad (14)$$

Eq. (14) is equivalent to Eq. (13) after removing constants. As a quadratic function, Eq. (14) bears a closed form solution in minimization according to the Parseval’s theorem after the Fourier transform. The optimal  $I$  is written as

$$\mathcal{F}(I) = \frac{\overline{\mathcal{F}(k)}\mathcal{F}(B + v) + \beta/\theta(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(w_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(w_y))}{\mathcal{F}(k)\mathcal{F}(k) + \beta/\theta(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(\partial_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(\partial_y))}. \quad (15)$$

The notations are the same as those in Eq. (6).



**Fig. 5.** Deconvolution result comparison. The blurred images in the top and bottom rows are with Gaussian and impulse noise respectively.

In solving for  $w$  and  $v$  given the  $I$  estimate, because  $w$  and  $v$  are not coupled with each other in the objective function (they belong to different terms), their optimization is independent. Two separate objective functions are thus yielded:

$$\begin{cases} E(w; I) = \frac{1}{2} \|w - \nabla I\|_2^2 + \theta \lambda \|w\|_2 \\ E(v; I) = \frac{1}{2} \|v - (I \otimes k - B)\|^2 + \beta \|v\| \end{cases} \quad (16)$$

Each objective function in Eq. (16) categorizes to a single-variable optimization problem because the variables for different pixels are not spatially coupled. The optimal solutions for all  $w_{x,s}$  can be derived according to the shrinkage formula:

$$w_x = \frac{\partial_x I}{\|\nabla I\|_2} \max(\|\nabla I\|_2 - \theta \lambda, 0). \quad (17)$$

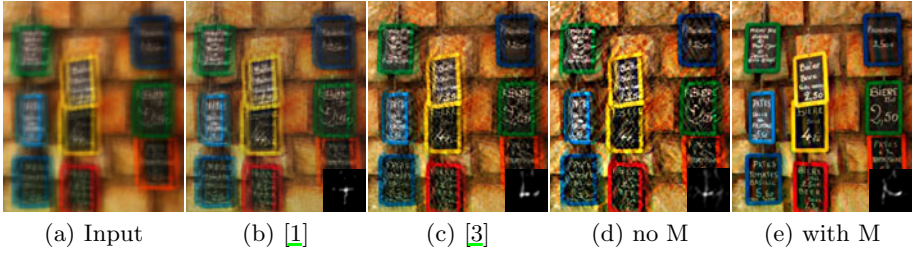
Here, isotropic TV regularizer is used – that is,  $\|\nabla I\|_2 = \sqrt{(\partial_x I)^2 + (\partial_y I)^2}$ .  $w_y$  can be computed similarly using the above method.

Computing  $v$  can be even simpler because it is an one-dimensional shrinkage:

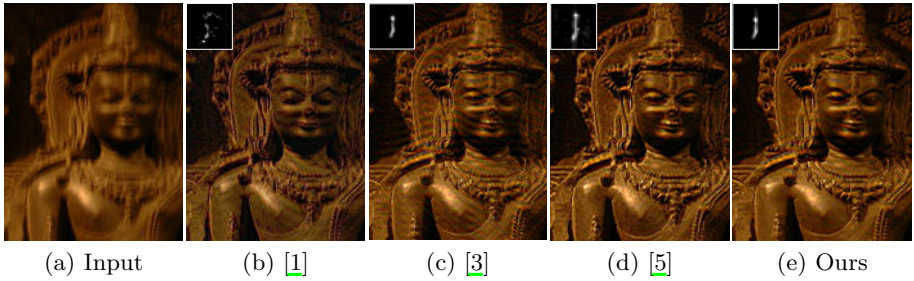
$$v = \text{sign}(I \otimes k - B) \max(\|I \otimes k - B\| - \beta, 0), \quad (18)$$

where  $\beta$  and  $\theta$  are two small positive values to enforce the similarity between the auxiliary variables and the respective terms. To further speed up the optimization, we employ the warm-start scheme [3, 16]. It first sets large penalties ( $\beta$  and  $\theta$  in our algorithm) and gradually decreases them in iterations. The details are shown in Alg. 3. We empirically set  $\beta_0 = 1$ ,  $\theta_0 = \lambda^{-1}$ , and  $\beta_{\min} = \theta_{\min} = 0.01$ .

Figure 5 shows examples where the blurred images are with Gaussian and impulse noise respectively. The TV- $\ell_1$  model performs comparably to other state-of-the-art deconvolution methods under the Gaussian noise. When significant impulse-like sensor noise exists, it works even better. In terms of the computation time, the methods of [15] and [17] spend 3 minutes and 1.5 seconds respectively to produce the results in Figure 5 with the provided implementation while our deconvolution algorithm, albeit using the highly non-linear function, uses 6s in Matlab. All methods deconvolve three color channels independently.



**Fig. 6.** Small objects such as the characters and thin frames are contained in the image. They greatly increase the difficulty of motion deblurring. (d)-(e) show our results using and not using the M map. The blur kernel is of size  $51 \times 51$ .



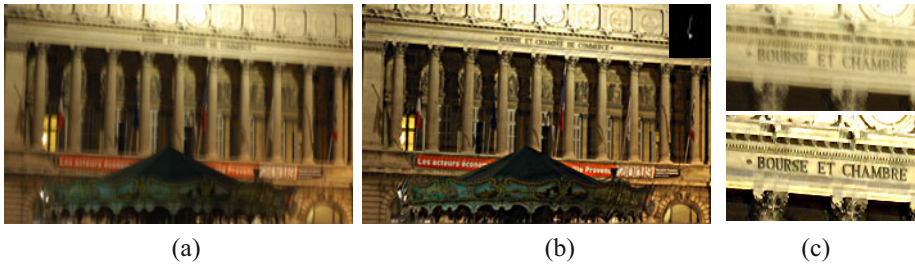
**Fig. 7.** Comparison of state-of-the-art deblurring methods

## 4 More Experimental Results

We experimented with several challenging examples where the images are blurred with large kernels. Our method generally allows using the default or automatically adapted parameter values. In the kernel estimation, we adaptively set the initial values of  $\tau_r$  and  $\tau_s$ , using the method of [5]. Specifically, the directions of image gradient are initially quantized into four groups.  $\tau_s$  is set to guarantee that at least  $2\sqrt{P_k}$  pixels participate in kernel estimation in each group, where  $P_k$  is the total number of pixels in kernel  $k$ .  $\tau_r$  is similarly determined by allowing at least  $0.5\sqrt{P_I P_k}$  pixels to be selected in each group.  $P_I$  is the total number of pixels in the input image. In the coarse kernel estimation phase, we set  $\lambda = 2e^{-3}$  and  $\gamma = 10$  to resist noise. In the kernel refinement, we set  $\gamma = 1$ .  $\lambda$  in the final image deconvolution is set to  $2e^{-2}$ .

Our two-phase kernel estimation is efficient because we put the non-convex optimization into the second phase. Our Matlab implementation spends about 25 seconds to estimate a  $25 \times 25$  kernel from an  $800 \times 600$  image with an Intel Core2Quad CPU@2.40G. The coarse kernel estimation uses 12s in the multi-scale framework while the kernel refinement spends 13s as it is performed only in the finest image scale.

In Figure 6(a), we show an example that contains many small but structurally-salient objects, such as the characters, which make high quality kernel estimation



**Fig. 8.** One more example. (a) Blurred image. (b) Our result. (c) Close-ups.

very challenging. The results (shown in (b) and (c)) of two other methods contain several visual artifacts due to imperfect kernel estimation. (d) shows our result without performing edge selection. Compared to the image shown in (e), its quality is lower, indicating the importance of incorporating the gradient mask  $M$  in defining the objective function.

Figure 7 shows another example with comparisons with three other blind deconvolution methods. The kernel estimates of Fergus *et al.* [1] and Shan *et al.* [3] are seemingly too sparse, due to the final hard thresholding operation. The restored image is therefore not very sharp. The deblurring result of Cho and Lee [5] contains some noise. Our restored image using Alg. 3 is shown in (e). We have also experimented with several other natural image examples. Figure 8 shows one taken under dim light. More of them are included in our supplementary file downloadable from the project website.

## 5 Concluding Remarks

We have presented a novel motion deblurring method and have made a number of contributions. We observed that motion deblurring could fail when considerable strong and yet narrow structures exist in the latent image and proposed an effective mask computation algorithm to adaptively select useful edges for kernel estimation. The ISD-based kernel refinement further improves the result quality with adaptive regularization. The final deconvolution step uses a  $\ell_1$  data term that is robust to noise. It is solved with a new iterative optimization scheme. We have extensively tested our algorithm, and found that it is able to deblur images with very large blur kernels, thanks to the use of the selective edge map.

## Acknowledgements

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 413110) and CUHK Direct Grant (No. 2050450).

## References

1. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Trans. Graph.* 25, 787–794 (2006)
2. Jia, J.: Single image motion deblurring using transparency. In: *CVPR* (2007)
3. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graph.* 27 (2008)
4. Joshi, N., Szeliski, R., Kriegman, D.J.: Psf estimation using sharp edge prediction. In: *CVPR* (2008)
5. Cho, S., Lee, S.: Fast motion deblurring. *ACM Trans. Graph.* 28 (2009)
6. Cai, J.F., Ji, H., Liu, C., Shen, Z.: Blind motion deblurring from a single image using sparse approximation. In: *CVPR*, pp. 104–111 (2009)
7. Joshi, N.: Enhancing photographs using content-specific image priors. PhD thesis, University of California, San Diego (2008)
8. You, Y., Kaveh, M.: Blind image restoration by anisotropic regularization. *IEEE Transactions on Image Processing* 8, 396–407 (1999)
9. Chan, T., Wong, C.: Total variation blind deconvolution. *IEEE Transactions on Image Processing* 7, 370–375 (1998)
10. Osher, S., Rudin, L.: Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis* 27, 919–940 (1990)
11. Alvarez, L., Mazon, L.: Signal and image restoration using shock filters and anisotropic diffusion. *SIAM J. Numer. Anal.* 31 (1994)
12. Gilboa, G., Sochen, N.A., Zeevi, Y.Y.: Regularized shock filters and complex diffusion. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2350, pp. 399–413. Springer, Heidelberg (2002)
13. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: *CVPR* (2009)
14. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.* 27 (2008)
15. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.* 26, 70 (2007)
16. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences* 1, 248–272 (2008)
17. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: *NIPS* (2009)
18. Yang, J., Zhang, Y., Yin, W.: An efficient TVL1 algorithm for deblurring multi-channel images corrupted by impulsive noise. *SIAM J. Sci. Comput.* 31, 2842–2865 (2009)
19. Bar, L., Sochen, N., Kiryati, N.: Image deblurring in the presence of salt-and-pepper noise. In: *International Conference on Scale Space and PDE methods in Computer Vision*, pp. 107–118 (2005)
20. Joshi, N., Zitnick, C.L., Szeliski, R., Kriegman, D.J.: Image deblurring and denoising using color priors. In: *CVPR*, pp. 1550–1557 (2009)
21. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.* 26, 1 (2007)
22. Wang, Y., Yin, W.: Compressed Sensing via Iterative Support Detection. CAAM Technical Report TR09-30 (2009)

# Single Image Deblurring Using Motion Density Functions

Ankit Gupta<sup>1</sup>, Neel Joshi<sup>2</sup>, C. Lawrence Zitnick<sup>2</sup>,  
Michael Cohen<sup>2</sup>, and Brian Curless<sup>1</sup>

<sup>1</sup> University of Washington

<sup>2</sup> Microsoft Research

**Abstract.** We present a novel single image deblurring method to estimate spatially non-uniform blur that results from camera shake. We use existing spatially invariant deconvolution methods in a local and robust way to compute initial estimates of the latent image. The camera motion is represented as a *Motion Density Function* (MDF) which records the fraction of time spent in each discretized portion of the space of all possible camera poses. Spatially varying blur kernels are derived directly from the MDF. We show that 6D camera motion is well approximated by 3 degrees of motion (in-plane translation and rotation) and analyze the scope of this approximation. We present results on both synthetic and captured data. Our system out-performs current approaches which make the assumption of spatially invariant blur.

## 1 Introduction

Image blur due to camera shake is a common problem in consumer-level photography. It arises when a long exposure is required and the camera is not held still. As the camera moves, the image formation process integrates a stream of photographs of the scene taken from slightly different viewpoints.

Removing blur due to camera shake is currently a very active area of research. Given only a single photograph, this blur removal is known as blind deconvolution, i.e., simultaneously recovering both the blur kernel and the deblurred, latent image. Commonly, it is assumed that the blur kernel is spatially invariant, reducing the set of camera motions that may be modeled.

An open problem is to model more general camera motions, which are quite common and can cause spatially varying blur. We focus on generalizing the camera motion to include both 2D translation and in-plane rotation. Thus, starting from a single image, we seek to recover the latent image, and the spatially varying blur kernels that arise from this more general camera motion.

We develop a novel formulation of the camera shake deblurring problem by generalizing spatially invariant (2D) kernels. Although a full model of motion would require 6 degrees of freedom, we show that for typical scenarios, 6D general motion can be reasonably approximated with a 3-dimensional motion (only in-plane rotation and translation). The problem is still substantially more under-constrained than the standard in-plane translation-only case.



Rather than directly recovering the spatially varying blur kernels at each image point, we observe that camera motion is a 1D curve through camera pose space. We model the time spent in each pose over the exposure as a density function in a higher dimensional camera motion space; we call this a *Motion Density Function* (MDF). The MDF can be used to generate the kernel at any location in the image without knowing the temporal ordering of the motion curve. Our system takes as input (1) a blurred image, (2) its EXIF tags specifying sensor resolution and approximate focal length, and (3) an estimate of the maximum blur kernel size, and recovers both the latent image and the MDF using a non-linear optimization scheme similar to a more traditional spatially invariant blind-deconvolution method. Altogether, we demonstrate an automatic method for single image deblurring under a range of spatially-varying, camera motion blurs.

The paper is organized as follows. In Section 2, we survey related work. In Sections 3 and 4, we propose and analyze our optimization formulation and then discuss our solution of this formulation in Section 5. In Section 6, we show the results of our approach and finally conclude with a discussion of limitations and future work in Section 7.

## 2 Related Work

Image deblurring has received a lot of attention in the computer vision community. Deblurring is the combination of two tightly coupled sub-problems: PSF estimation and non-blind image deconvolution. These problems have been addressed both independently and jointly [1]. Both are longstanding problems in computer graphics, computer vision, and image processing.

Image blur arises from multiple causes. Image blur due to camera motion has recently received increased attention, as it is a very common problem in consumer-level photography. In most recent work, image blur is modeled as the convolution of an unobserved latent image with a single, spatially invariant blur kernel [1,2,3,4,5,6,7,8,9,10,11].

Software-based methods use image priors and kernel priors to constrain an optimization for the blur kernel and the latent image [2,3,4,5,6,12,13,14].

Fergus et al. [4] recover a blur kernel by using a natural image prior on image gradients in a variational Bayes framework. Shan et al. [2] incorporate spatial parameters to enforce natural image statistics using a local ringing suppression step. Jia et al. [13] use transparency maps to get cues for object motion to recover blur kernels by performing blind-deconvolution on the alpha matte, with a prior on the alpha-matte. Joshi et al. [14] predict a sharp image that is consistent with an observed blurred image. They then solve for the 2D kernel that maps the blurred image to the predicted image.

Levin et al. [15] give a nice overview of several of these existing deblurring techniques. Common to all of them is that they assume spatial invariance for the blur. Levin et al. show that spatial invariance is often violated, as it is only valid in limited cases of camera motion. Their experiments show that in practice in-plane camera rotation (i.e., roll), which leads to spatially varying blur kernels, is quite common.

There is relatively little work on handling spatially-varying blur. Tai et al. [16] developed a hybrid camera which captured a high frame rate video and a blurred image. Optical flow vectors from the video are used to guide the computation of spatially-varying blur kernels which are in turn used for deblurring. This method is limited by the requirement of a hybrid camera and faces problems in regions where optical flow computation fails. Tai et al. [17] use a coded exposure to produce a stroboscopic motion image and estimate motion homographies for the discrete motion steps with some user interaction, which are then used for deblurring. Their method requires close user interaction and relies on non-overlapping texture information in the blurred regions. Dai et al. [18] propose a method to estimate spatially varying blur kernels based on values of the alpha map. The method relies strongly on the pre-computation of a good alpha matte and assumes the scene to be a foreground object moving across a background. Shan et al. [19] propose a technique to handle rotational motion blur. They require user interaction for rotation cues and also rely on constraints from the alpha matte.

One approach to model the spatial variation of blur kernels is to run a blind deconvolution method at each pixel. Joshi et al. [14] do this in a limited sense, where they run their method for non-overlapping windows in an image and use this to remove spatially varying defocus blur and chromatic aberration; however, they do not address camera motion blur, nor do they try to recover a global model of the blur. Levin et al. [12] take a similar approach for object motion blur, where an image is segmented into several areas of different motion blur and then each area is deblurred independently. Hirsch et al. [20] also propose a multi-frame patch-based deblurring approach but do not impose any global camera motion constraints on the spatially-varying blur.

Unfortunately, these approaches have several limitations. First, running blind deconvolution for each pixel, window, or segment can be slow. Furthermore, it is unclear how best to handle boundaries between areas with different blur kernels, which can lead to artifacts. Second, deblurring techniques often use natural image priors, which is inherently a global constraint, and may not apply to all local areas in an image, thus leading to unreliable blur kernels and artifacts in the deblurred result.

In comparison, we do not try to recover the spatially varying blur kernels directly, but rather recover the camera motion (specifically the MDF) from which the blur kernels can be derived. In a concurrent work, Whyte et al. [21] describe a similar framework where they recover 3-dimensional rotational camera motion (roll, pitch, and yaw) to explain the spatially-varying blur. In contrast, we recover a different set of 3D camera motions (roll and x,y-translations). Our results show that these two approaches are similar for sufficiently long focal lengths due to the rotation-translation ambiguity in that focal length range. However at shorter focal lengths, each system will result in different types of artifacts depending on the errors in approximating the actual underlying camera motion. Thus, the two papers taken together form a nicely complementary set of results. We present a more detailed analysis of this rotation-translation ambiguity in Section 4.

### 3 A Unified Camera Shake Blur Model

In this section, we develop a unified model relating the camera motion, the latent image and the blurred image for a scene with constant depth.

#### 3.1 Image Blur Model

Let  $l$  be the latent image of a constant depth scene and  $b$  be the recorded blurred image. The blurred image can be written as a convolution of the latent image with a kernel  $k$  and the addition of some noise  $n$ . The convolution model does not account for variations in depth and view-dependent illumination changes and we do not handle them here:

$$b = k \otimes l + n, \quad (1)$$

For simplicity, we assume Gaussian noise,  $n \sim \mathcal{N}(0, \sigma^2)$ .

This convolution model can also be written as a matrix-vector product:

$$B = \mathcal{K}L + N, \quad (2)$$

where  $L$ ,  $B$ , and  $N$  denote the column-vector forms of  $l$ ,  $b$ , and  $n$  respectively.  $\mathcal{K}$  is an image filtering matrix that applies the convolution – each row of  $\mathcal{K}$  is the blur kernel placed at each pixel location and unraveled into a row vector. For this reason, we also refer to  $\mathcal{K}$  as the blur matrix. With spatially invariant blur each row has the same values that are just shifted in location. This matrix-vector form becomes particularly useful for formulating spatially varying blur – as each row contains a different blur kernel for each pixel [22], as we will discuss in the next section.

#### 3.2 Blur Matrix as Motion Response

We assume the camera initially lies at the world origin with its axes aligned with the world axes. A camera motion is a sequence of camera poses where each pose can be characterized by 6 parameters - 3 rotations and 3 translations. Any camera motion can be represented as a 1D continuous path through this 6-dimensional space, which we call *camera pose space*. In a discretized version of this space, the camera spends a fraction of the exposure time at each pose; we call this proportion the *density* at that pose. Taken all together, these densities form a Motion Density Function from which a blur kernel can be directly determined for any point on the image. The MDF for all the camera poses forms a column vector over the discrete positions in the camera pose space. We denote the MDF by  $A$  where each element  $a_j$  denotes the density at the camera pose  $j$ .

The observed blurred image  $B$  is an integration over the images seen by the camera over all the poses in its path. In the discrete motion space,  $B$  is a summation over the images seen by the camera in all possible poses, each weighted by

the proportion of time spent by the camera in that pose, which in our notation is the pose’s *density*. We write this mathematically as:

$$B = \sum_j a_j(K_j L) + N, \quad (3)$$

where  $K_j$  is a matrix that warps  $L$ , the latent image or the un-blurred image seen by the camera in the original pose, to the image seen in pose  $j$ .  $N$  is the noise model introduced in Section 3.1. Given a particular 6D pose (indexed by  $j$ ) of a camera, we denote the corresponding homography that warps a fronto-parallel scene at depth  $d$  as  $P_j$ :

$$P_j = C(R_j + \frac{1}{d}t_j[0 \ 0 \ 1])C^{-1}, \quad (4)$$

where  $R_j$  and  $t_j$  are the rotation matrix and translation vector for pose  $j$  and  $C$  is the matrix of camera intrinsics, which we form from the information in the image EXIF tags. For now we assume the depth  $d$  is known.  $K_j$  is an image warping matrix where each row contains the weights used to compute the values of pixels in the warped image by applying the inverse homography. We use bilinear interpolation for the warps and thus there are at most four non-zero values per row of  $K_j$ . For clarity, we note that  $K_j$  is a square matrix where each dimension is the width times the height of the image  $l$ .

Rearranging the linear operations in Equation 3 and comparing it with Equation 2, allows us to write the blur matrix  $\mathcal{K}$  as:

$$\mathcal{K} = \sum_j a_j K_j. \quad (5)$$

Thus the  $K_j$ ’s form a basis set whose elements can be linearly combined using the MDF to get the corresponding blur matrix for any camera path. By definition, the blur matrix also gives us the blur kernels for each pixel location in the image. We call this basis set the *Motion Response Basis* (MRB). We note that the MRB can represent any basis in the more traditional sense, e.g., each basis matrix could actually correspond to an aggregate blur matrix itself where it captures some region of support in the 6D space.

In this work, we choose a particular basis that we found meaningful for modeling typically occurring camera motion blurs. Specifically, we choose to reduce motion in the 6D space to a 3D subspace: rotation around the  $z$  axis (*roll*) and  $x$  and  $y$  translation (modeling  $x$  translation and *yaw* and  $y$  translation and *pitch* together, respectively and neglecting the affect on  $z$  translation). We then compute the basis by point-sampling this 3D space. We discuss the validity of using a 3D space and details about creating basis sets in Section 4.

### 3.3 Optimization Formulation

Equation 3 relates the MDF to the latent image and the blurred image. In the process of deblurring, each basis matrix  $K_j$  is pre-computed and we solve for

the variables  $L$  and  $A$ . To do this we pose the problem in a Bayesian framework and seek to recover the latent image and MDF that is most likely given the observation and priors on the image and MDF.

We compute the maximum a posteriori (MAP) estimate, which we formulate as the minimization of the following energy function (in the interest of space, we have left out the intermediate derivation steps from the Bayesian formulation):

$$E = \|\sum_j a_j K_j L - B\|^2 + \text{prior}(L) + \text{prior}(A), \quad (6)$$

$$\text{prior}(L) = \phi(|\partial_x L|) + \phi(|\partial_y L|), \quad (7)$$

$$\text{prior}(A) = \lambda_1 \|A\|^\gamma + \lambda_2 \|\nabla A\|^2. \quad (8)$$

$\text{prior}(L)$  is the global image prior with the same parameter settings as used by Shan et al. [2].  $\phi$  assigns a linear penalty to small gradients and quadratic penalties to large gradients and approximates the heavy-tailed gradient distribution priors for natural images [23].

$\text{prior}(A)$  models priors that are important to recovering an accurate MDF. Specifically in 6D, the camera motion is a 1D path that captures the trajectory that the camera takes during the exposure window. This holds in the 3D space as well. Ideally, one would enforce a path prior directly on the MDF; however, this is a computationally challenging constraint to optimize. Thus we enforce two other computationally more tractable constraints.

The first component of  $\text{prior}(A)$  is a sparsity prior on the MDF values. We note that while blur kernels in the 2D image space may seem quite dense, in the higher dimensional MDF space, a 1D path represents an extremely sparse population of the space. The second component of  $\text{prior}(A)$  is a smoothness prior on the MDF, which also incorporates the concept of the MDF representing a path, as it enforces continuity in the space and captures the conditional probability that a particular pose is more likely if a nearby pose is likely.

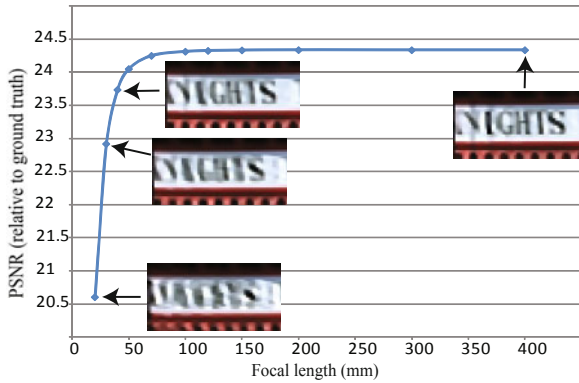
We also note that we can choose to use the whole blurred image for the optimization or some selected parts by masking out rows in  $L$ ,  $B$ , and the corresponding matrices.

## 4 Forming the Motion Response Basis

As introduced in Section 3, the Motion Response Basis (MRB) is the set of image warping matrices  $K_j$ 's that correspond to a warp operation relating the image in the original camera pose to that in camera pose  $j$ . We can pre-compute the MRB; however, the size of the basis set is critical for the computational feasibility of the system. We now discuss the issues involved in this computation.

### 4.1 Dependence on Scene Depth

As discussed in Section 3.2, it is necessary to know the scene depth  $d$  to compute the homographies  $P_j$  and corresponding basis matrices  $K_j$ . Unfortunately, recovering the depth of a scene from a single image is an under-constrained problem.

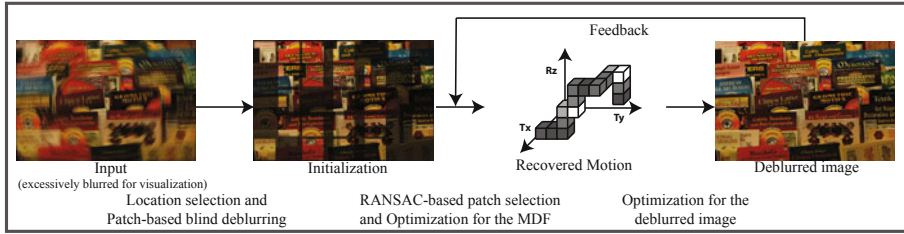


**Fig. 1.** PSNR of the deblurred result (by recovering *yaw* using *x* translation) with respect to the ground truth image. Cropped parts from deblurred images for some plot samples are also shown for qualitative comparison.

Fortunately, given our assumption of a constant depth or fronto-parallel scene, we observe that we do not need to know the exact depth and rather can consider  $\frac{1}{d}t_j$  as a single 3-dimensional variable, which allows us to remove the dependence on depth and instead only concern ourselves with the image parallax. Given this, the number of degrees of freedom of our system does not change, depth is not needed as a separate variable. Computing a basis that captures the parallax reduces to sampling the total resulting image plane translation, appropriately. We discuss how to choose the sampling resolution in Subsection 4.3.

## 4.2 Computational Reduction in d.o.f for the Camera Motion

We observe that instead of using 6 degrees of freedom for the camera motion, we can use only 3 degrees of freedom - *roll* (rotation about *z*-axis) and *x* and *y* translations. This reduction makes the problem computationally more feasible since the size of the MRB is dependent on the number of degrees of freedom. Given the projective camera model, it is known that small camera rotations about the *x* (*pitch*) and *y* (*yaw*) axes can be approximated by camera translations when perspective affects are minimal (i.e., longer focal lengths). Joshi et al [24] show that in most cases the camera shake motion lies in this operating range. To validate this approximation, we performed an experiment with a ground truth image blurred using a synthetic camera motion involving *yaw*. We then solve for an MDF limited to only *x* translations. Figure 1 shows the PSNR values comparing the resulting deconvolved images to the ground truth as we vary the focal length. We varied the amount of *yaw* in relation to focal length to keep the blur kernels approximately the same size ( $\sim 11$  pixels wide) so that deconvolution errors due to differences in blur kernel size do not affect the analysis. The PSNR improvement levels out quickly, which means that the recovered translations



**Fig. 2.** Our System Pipeline

start to accurately approximate the *yaw* as the focal length increases to a value that covers most standard camera settings. A similar argument also holds for *pitch* to be approximated by *y* translations. We do a similar analysis for the effect of *z* translations of the camera and found that their contribution is also negligible under typical camera shake motion. We provide more analysis including similar figures for *pitch* and *z* translations on the project webpage [25].

As a result, the full 6D space of camera motion can be accurately approximated using only samples of the 3D space of *roll*, *x* and *y* translations across a wide range of focal lengths. We note that 6D motions can still be theoretically solved using our framework, but the high dimensionality makes the solution computationally prohibitive.

### 4.3 Sampling Range and Resolution of the Camera Motion Space

The number of matrices  $K_j$  is the number of motion poses that we sample. The number of samples along each d.o.f. affects the size of the MRB and hence we want to keep the sampling as coarse as possible. We hypothesize that the sampling needs to only be dense enough that the neighboring voxels in the discretized motion space project to within a pixel width at any image location. The range of the motion can be chosen to cover the estimate of the kernel size that is initially specified by the user. Hence we automatically choose the sampling range and resolution along the 3 degrees of freedom and pre-compute the  $K_j$ 's.

## 5 Our System

The proposed optimization in Equation 6 is non-linear in the variables  $L$  and  $A$ . We solve this using an alternating, iterative EM-style procedure which takes an initialization for  $L$  and then optimizes for  $A$  and  $L$  successively. Figure 2 shows the steps in our system pipeline, and we explain each of the steps now.

### 5.1 Generating an Initial Estimate for the Latent Image

We first select uniformly distributed patches on the blurred image which we independently deblur for generating an initial estimate for the latent image  $L$ .

The patch sizes are proportional to the estimated maximum blur kernel size in the image, which is an input parameter for our system. Since kernel estimation techniques require a good distribution of edge orientations [14], we filter out the patches having a low average value of the Harris corner metric. The Harris corner metric measures the presence of gradients in orthogonal directions in an image region and hence is a good estimate for the success of kernel estimation. We empirically choose this threshold value to be 0.1. We denote these selected patches as  $p_i$ 's. We then use the blind deconvolution approach proposed by Shan et al [2] to deblur each of these patches independently. We denote the corresponding deblurred patches and blur kernels as  $d_i$ 's and  $k_i$ 's, respectively. These deblurred patches are the initial estimates for the latent image in corresponding regions.

## 5.2 Ransac-Based Optimization for the MDF

Assuming we know  $L$ , we can solve for  $A$  by minimizing the following function which is a reduced form of Equation 6

$$E = \left\| \sum_j a_j (K_j L) - B \right\|^2 + \lambda_1 \|A\|^\gamma + \lambda_2 \|\nabla A\|^2 \quad (9)$$

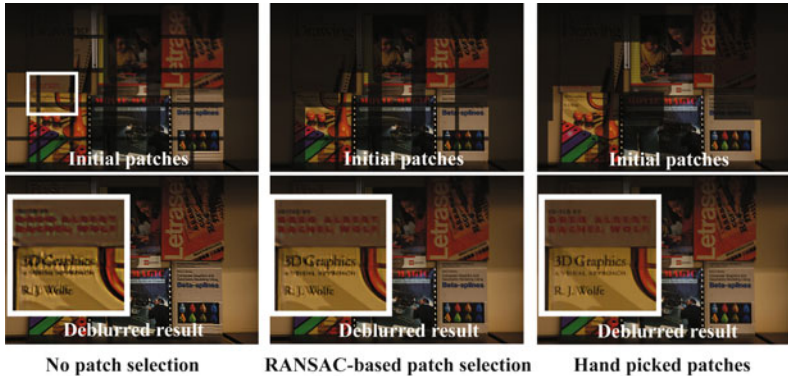
Here we only use the parts of the image regions of  $L$  and  $B$  which are covered by patches  $p_i$ 's. This optimization is performed using an iterative re-weighted least squares (IRLS). We use the values of  $\gamma = 0.8$ ,  $\lambda_1 = 0.1$  and  $\lambda_2 = 0.5$  in all our experiments. In practice, we see that using five iterations of IRLS works well.

We have found that using all the deblurred patches from the initialization phase does not give good results. This is because blind blur kernel estimation on a patch can vary in performance based on the quality and quantity of texture information and image noise. Ideally, we would want to select the best deblurred patches in the image for fitting the MDF. Unfortunately, this is a hard problem to solve in itself. There are numerous metrics that have been used in the literature for this classification – penalizing based on a heavy-tailed gradient distribution ([23]) and slope of the power spectrum ([26]); however, we have not found these to work well.

Instead, we use a RANSAC-based scheme to robustly choose a set of “good” patches from the initial set of deblurred patches. Each RANSAC iteration randomly chooses 40% of the patches and fits an MDF to them by minimizing Equation 9. We classify each of the patches,  $p_i$ 's, as inliers or outliers by how well the MDF describes the corresponding blur kernel. We consider this a contribution of using an MDF – the process of fitting a lower-dimensional MDF to blurred/deblurred patch pairs allows us to measure the quality of deblurring in local image patches, which is otherwise difficult.

Specifically, to compute the inlier metric, let  $k'_i$  be the recovered kernel using the MDF, the residual error is given as,  $\|d_i * k'_i - b_i\|_2$ . A patch is an inlier if its residual error is less than 1.2 times the median of all the errors in the patch set. From all the RANSAC iterations, we select the set of patches which gives the minimum average residual error on the inliers. Finally, we fit an MDF using all the inlier patches.





**Fig. 3.** Deblurred results using different schemes for choosing good deblurred patches for initialization. Handpicking patches works better than RANSAC-based selection which works better than no selection at all.

To test the performance of our RANSAC approach, we ran our complete deblurring pipeline on three cases – (A) using all the initially deblurred image patches, (B) automatically choose inliers from the initially deblurred patches using RANSAC, and (C) handpicking patches to be deblurred. Figure 3 shows the deblurring results for these three cases, and we see that using handpicked patches works better than RANSAC which in turn works better than doing no patch selection. We use the RANSAC-based scheme in all our experiments since it is robust, automatic, and gives reasonable results.

### 5.3 Optimization for the Latent Image

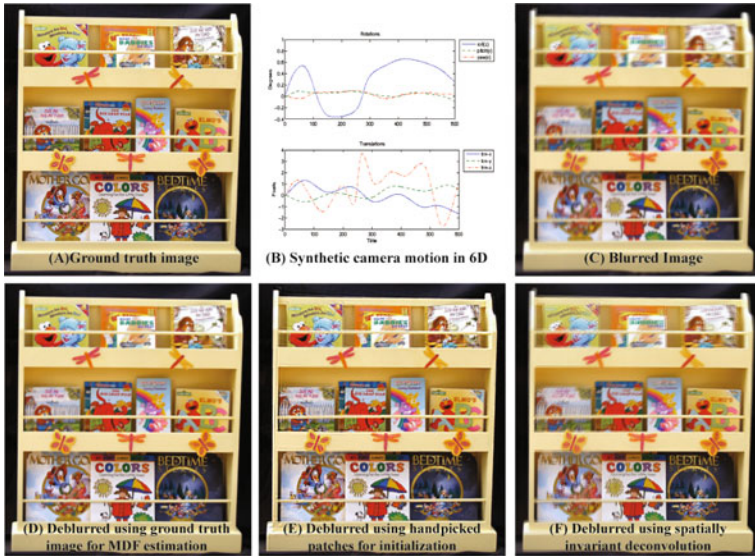
Assuming we know the MDF  $A$ , we can solve for  $L$  by minimizing the following function which is another reduced form of Equation 6. This is essentially a non-blind image deconvolution:

$$E = \left\| \sum_j (a_j K_j) L - B \right\|^2 + \phi(|\partial_x L|) + \phi(|\partial_y L|). \quad (10)$$

We solve this optimization as described by Shan et al. [23] in their paper. We feed the solution back into the RANSAC-based MDF optimization and repeat the overall procedure until the latent image converges. We have observed that 2-3 iterations are enough for convergence of the recovered latent image in all our experiments.

## 6 Experiments and Results

We run our deblurring experiments on a quad dual-core 3.0GHz PC with 64GB RAM. Running our system on a 768X512 sized image takes about 1 hour and

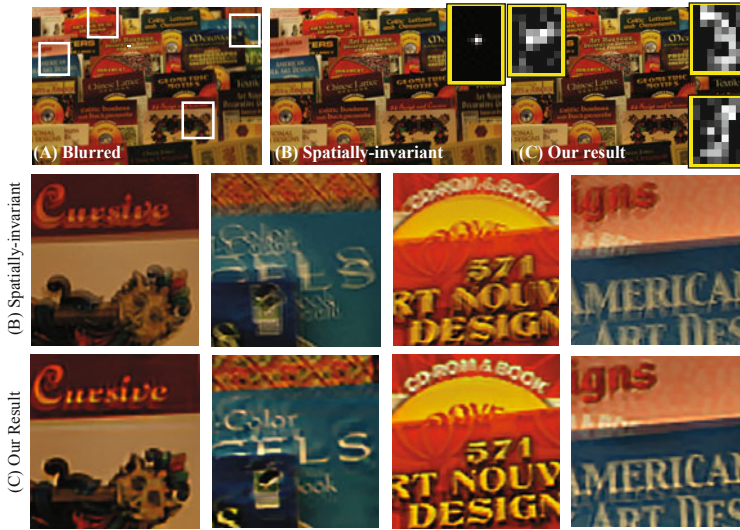


**Fig. 4.** Model validation. (A) Ground truth image, (B) Synthetic camera motion, (C) Blurred image, (D) Deblurred output using ground truth initialization, (E) Deblurred output using handpicked locally blind deblurred image regions for initialization, (F) Result with a globally spatially invariant deblurring system. (Please zoom in to compare.)

takes up around 8 GBs of RAM. As there are no existing single image automatic deblurring systems for a general (spatially-varying) camera motion blur, we perform all our comparisons with the recent blind deconvolution method of Shan et al [2], who have code available online.

## 6.1 Model Validation Using Synthetic Data

Figure 4 shows the visual validation of our model formulation and illustrates sensitivity to the initialization values. We take a sharp image (A) and use a specified 6D camera motion (B) to blur it (C). We then optimize for the 3D MDF, ( $z$ -rotation and  $x$ ,  $y$ -translations) using the original sharp image as the initialization (D) and using some handpicked locally blind deblurred image regions for initialization (E). (F) shows the corresponding result for a blind spatially invariant deblurring system. We see that (D) is very close to the ground truth which means that if we start with the ideal initialization, we can recover a very accurate 3D approximation of the true 6D MDF. We do see some artifacts (upper left corner) in (E) which shows that our system is sensitive to the initial latent image. But we still out-perform the result in (F), which assumes only a translational motion of the camera or in other words, a spatially invariant blur kernel. We show more such comparison sets with synthetic data on the project webpage [25].



**Fig. 5.** Deblurring result. (A) Blurred image, (B) Deblurred image using spatially invariant approach, (C) Deblurred result using our system. Recovered blur kernels at few locations are shown in yellow boxes.



**Fig. 6.** Deblurring result. (A) Blurred image, (B) Deblurred image using spatially invariant approach, (C) Deblurred result using our system. Recovered blur kernels at few locations are shown in yellow boxes.

## 6.2 Results and Comparisons on Real-World Data

Figure 5 shows one of our results for real-world blurred images of scenes captured using a Canon EOS-1D camera. It shows the original blurred image (A), the deblurred result using spatially invariant deconvolution (B), our deblurred result (C), and the inset comparisons between (B) and (C). Our approach shows a significant improvement over the spatially invariant approach in all the cases. Our current implementation does not handle depth variance in the scene. Figure 6 is a difficult example as it has large depth variation, yet our deblurring method performs better than the spatially invariant approach and gives a reasonable looking deblurred result. This shows that our system can handle depth variation until the point where it starts to cause a large amount of spatial variation in the blur kernels. We also provide more results and intermediate step images for each of these results on the project webpage 25.

## 7 Discussion and Future Work

We presented a unified model of camera shake blur and a framework to recover the camera motion and latent image from a single blurred image. One limitation of our work is that it depends on imperfect spatially invariant deblurring estimates for initialization. Two things could improve this: (a) using other blur estimation methods for initialization and (b) a better metric to judge the accuracy of a particular kernel estimate, which is still a very open and interesting problem.

Another interesting area of work is to explore other motion response bases. Instead of using a uniform sampling with delta functions, a more sophisticated basis with larger, more complex support regions may be more appropriate for modeling common camera blurs.

## Acknowledgements

We would like to thank the reviewers for their insightful comments. We would also like to thank Qi Shan for useful discussions regarding blind/non-blind image deblurring methods. This work was supported by funding from the University of Washington Animation Research Labs, Microsoft, Adobe, and Pixar.

## References

1. Richardson, W.H.: Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America* (1917-1983) (1972)
2. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. In: *ACM SIGGRAPH 2008 Papers* (2008)
3. Likas, A.C., Galatsanos, N.P.: A variational approach for bayesian blind image deconvolution (2004)

4. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. In: ACM SIGGRAPH 2006 Papers (2006)
5. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. In: ACM SIGGRAPH 2007 Papers (2007)
6. Joshi, N., Zitnick, L., Szeliski, R., Kriegman, D.: Image deblurring and denoising using color priors. In: Proceedings of IEEE CVPR '09 (2009)
7. Ben-Ezra, M., Nayar, S.K.: Motion-based motion deblurring. IEEE Trans. Pattern Analysis Machine Intelligence (2004)
8. Levin, A., Sand, P., Cho, T.S., Durand, F., Freeman, W.T.: Motion-invariant photography. In: ACM SIGGRAPH 2008 Papers (2008)
9. Agarwal, A., Xu, Y.: Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In: Proceedings of IEEE CVPR '09 (2009)
10. Agarwal, A., Xu, Y., Raskar, R.: Invertible motion blur in video. In: ACM SIGGRAPH 2009 Papers (2009)
11. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. In: ACM SIGGRAPH 2006 Papers (2006)
12. Levin, A.: Blind motion deblurring using image statistics. In: Advances in Neural Information Processing Systems (2007)
13. Jia, J.: Single image motion deblurring using transparency. In: Proceedings of IEEE CVPR '07, pp. 1–8 (2007)
14. Joshi, N., Szeliski, R., Kriegman, D.J.: Psf estimation using sharp edge prediction. In: Proceedings of IEEE CVPR '08 (2008)
15. Levin, A., Weiss, Y., Durand, F.: Understanding and evaluating blind deconvolution algorithms. In: Proceedings of IEEE CVPR '09 (2009)
16. Tai, Y.W., Du, H., Brown, M., Lin, S.: Image/video deblurring using a hybrid camera. In: Proceedings of IEEE CVPR '08 (2008)
17. Tai, Y.W., Kong, N., Lin, S., Shin, S.Y.: Coded exposure imaging for projective motion deblurring. In: Proceedings of IEEE CVPR '10 (2010)
18. Dai, S., Wu, Y.: Motion from blur. In: Proceedings of IEEE CVPR '08 (2008)
19. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: Proceedings of IEEE ICCV '07 (2007)
20. Hirsch, M., Sra, S., Schölkopf, B., Harmeling, S.: Efficient filter flow for space-variant multiframe blind deconvolution. In: Proceedings of IEEE CVPR '10 (2010)
21. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: Proceedings of IEEE CVPR '10 (2010)
22. Seitz, S., Winder, S.: Filter flow. In: Proceedings of IEEE ICCV '09 (2009)
23. Weiss, Y., Freeman, W.T.: What makes a good model of natural images? In: Proceedings of IEEE CVPR '07 (2007)
24. Joshi, N., Kang, S.B., Zitnick, C.L., Szeliski, R.: Image deblurring using inertial measurement sensors. In: ACM SIGGRAPH 2007 Papers (2010)
25. Gupta, A.: Project webpage: Single image deblurring using motion density functions (2010), [http://www.grail.cs.washington.edu/projects/mdf\\_deblurring](http://www.grail.cs.washington.edu/projects/mdf_deblurring)
26. Reinhard, E., Shirley, P., Ashikhmin, M., Troscianko, T.: Second order image statistics in computer graphics. In: Proceedings of the ACM Symposium on Applied Perception in Graphics and Visualization (2004)

# An Iterative Method with General Convex Fidelity Term for Image Restoration\*

Miyoung Jung, Elena Resmerita, and Luminita Vese

Department of Mathematics, University of California, Los Angeles, U.S.A.  
Industrial Mathematics Institute, Johannes Kepler University, Linz, Austria

**Abstract.** We propose a convergent iterative regularization procedure based on the square of a dual norm for image restoration with general (quadratic or non-quadratic) convex fidelity terms. Convergent iterative regularization methods have been employed for image deblurring or denoising in the presence of Gaussian noise, which use  $L^2$  [1] and  $L^1$  [2] fidelity terms. Iusem-Resmerita [3] proposed a proximal point method using inexact Bregman distance for minimizing a general convex function defined on a general non-reflexive Banach space which is the dual of a separable Banach space. Based on this, we investigate several approaches for image restoration (denoising-deblurring) with different types of noise. We test the behavior of proposed algorithms on synthetic and real images. We compare the results with other state-of-the-art iterative procedures as well as the corresponding existing one-step gradient descent implementations. The numerical experiments indicate that the iterative procedure yields high quality reconstructions and superior results to those obtained by one-step gradient descent and similar with other iterative methods.

**Keywords:** proximal point method, iterative regularization procedure, image restoration, bounded variation, Gaussian noise, Laplacian noise.

## 1 Introduction

Proximal point methods have been employed to stabilize ill-posed problems in infinite dimensional settings, using  $L^2$  [1] and  $L^1$  data-fitting terms [2], respectively. Recently, Iusem-Resmerita [3] proposed a proximal point method for minimizing a general convex function defined on a general non-reflexive Banach space which is the dual of a separable Banach space. Our aim here is to investigate and propose, based on that method, several iterative approaches for image restoration.

In Tadmor et al [4], an iterative procedure for computing hierarchical  $(BV, L^2)$  decompositions has been proposed for image denoising, and this was extended to more general cases for image restoration and segmentation in [5]. Osher et al [1]

---

\* This research has been supported in part by a UC Dissertation Year Fellowship, by Austrian Science Fund Elise Richter Scholarship V82-N18 FWF, and by NSF-DMS grant 0714945.

proposed another iterative procedure for approximating minimizers of quadratic objective functions, with the aim of image denoising or deblurring, providing significant improvements over the standard model introduced by Rudin, Osher, Fatemi (ROF) [6]. This turned out to be equivalent to proximal point algorithm on a nonreflexive Banach space as well as to an augmented Lagrangian method for a convex minimization problem subject to linear constraints. In addition, He et al [2] generalized the Bregman distance based iterative algorithm [1] to  $L^1$  fidelity term by using a suitable sequence of penalty parameters, and proved the well-definedness and the convergence of the algorithm with  $L^1$  fidelity term, which is an iterative version of  $L^1$ -TV considered by Chan and Esedoglu [7], and presented denoising results in the presence of Gaussian noise. Benning and Burger [8] derived basic error estimates in the symmetric Bregman distance between the exact solution and the estimated solution satisfying an optimality condition, for general convex variational regularization methods. Furthermore, the authors of [8] investigated specific error estimates for data fidelity terms corresponding to noise models from imaging, such as Gaussian, Laplacian, Poisson, and multiplicative noise.

Recently, Iusem and Resmerita [3] combine the idea of [1] with a surjectivity result, shown in [9] and [10], in order to obtain a proximal point method for minimizing more general convex functions, with interesting convergence properties. For the optimization case where the objective function is not necessarily quadratic, they use a positive multiple of an inexact Bregman distance associated with the square of the norm as regularizing term; a solution is approached by a sequence of approximate minimizers of an auxiliary problem. Regarding the condition of being the dual of a Banach space, we recall that nonreflexive Banach spaces which are duals of other spaces include the cases of  $l_\infty$  and  $L^\infty$ ,  $l_1$  and  $BV$  (the space of functions of bounded variation) which appear quite frequently in a large range of applications.

Here we apply the proximal point method introduced in [3] to general ill-posed operator equations and we propose several algorithms for image restoration problems, such as image deblurring in the presence of noise (for Gaussian or Laplacian noise with corresponding convex fidelity terms). Finally, numerical results are given for each image restoration model. Comparisons with other methods of similar spirit or one-step gradient descent models are also presented.

## 2 Proposed Iterative Method for Solving Ill-Posed Operator Equations

Our proposed iterative method is based on the proximal point method and convergence results of Iusem-Resmerita from [3]. We first recall the necessary definitions and terminology. Let  $X$  be a nonreflexive Banach space and  $X^*$  its topological dual. For  $u^* \in X^*$  and  $u \in X$ , we denote by  $\langle u^*, u \rangle = u^*(u)$  the duality pairing. Denote by  $h(u) = \frac{1}{2}\|u\|^2$ , for  $u \in X$ .

For  $\varepsilon > 0$ , the  $\varepsilon$ -subdifferential of  $h$  at a point  $u \in X$  is [11]

$$\partial_\varepsilon h(u) = \{u^* \in X^* : h(v) - h(u) - \langle u^*, v - u \rangle \geq -\varepsilon, \forall v \in X\}.$$

The normalized  $\varepsilon$ -duality mapping of  $X$ , introduced by Gossez [9], extends the notion of duality mapping as follows

$$J_\varepsilon(u) = \{u^* \in X^* : \langle u^*, u \rangle + \varepsilon \geq \frac{1}{2}\|u^*\|^2 + \frac{1}{2}\|u\|^2\}, \tag{1}$$

and an equivalent definition for the  $\varepsilon$ -duality mapping is  $J_\varepsilon(u) = \partial_\varepsilon \left(\frac{1}{2}\|u\|^2\right)$ .

The inexact Bregman distances with respect to the convex function  $h$  and to an  $\varepsilon$ -subgradient  $\xi$  of  $h$  were defined in [3] as follows:

$$D^\varepsilon(v, u) = h(v) - h(u) - \langle \xi, v - u \rangle + \varepsilon. \tag{2}$$

Note that  $D^\varepsilon(v, u) \geq 0$  for any  $u, v \in X$  and  $D^\varepsilon(u, u) = \varepsilon > 0$  for all  $u \in X$ .

Given  $\varepsilon \geq 0$  and a function  $g : X \rightarrow \mathbb{R} \cup \{+\infty\}$ , we say that  $\bar{u} \in \text{dom } g = \{u \in X : g(u) < \infty\}$  is an  $\varepsilon$ -minimizer of  $g$  when

$$g(\bar{u}) \leq g(u) + \varepsilon \tag{3}$$

for all  $u \in \text{dom } g$ .

The following proximal point algorithm is proposed in [3] by Iusem and Resmerita.

**Initialization**

Take  $u_0 \in \text{dom } g$  and  $\xi_0 \in J_{\varepsilon_0}(u_0)$ .

**Iterative step**

Let  $k \in \mathbb{N}$ . Assume that  $u_k \in \text{dom } g$  and  $\xi_k \in J_{\varepsilon_k}(u_k)$  are given. We proceed to define  $u_{k+1}, \xi_{k+1}$ . Define  $D^{\varepsilon_k}(u, u_k) = h(u) - h(u_k) - \langle \xi_k, u - u_k \rangle + \varepsilon_k$  and  $\bar{\varepsilon}_k = \lambda_k \varepsilon_{k+1}$ .

Determine  $u_{k+1} \in \text{dom } g$  as an  $\bar{\varepsilon}_k$ -minimizer of the function  $g_k(u)$  defined as

$$g_k(u) = g(u) + \lambda_k D^{\varepsilon_k}(u, u_k), \tag{4}$$

that is to say, in view of (3),

$$g(u_{k+1}) + \lambda_k D^{\varepsilon_k}(u_{k+1}, u_k) \leq g(u) + \lambda_k D^{\varepsilon_k}(u, u_k) + \bar{\varepsilon}_k \tag{5}$$

for all  $u \in \text{dom } g$ .

Let  $\eta_{k+1} \in \partial g(u_{k+1})$  and  $\xi_{k+1} \in J_{\varepsilon_{k+1}}(u_{k+1})$  such that

$$\eta_{k+1} + \lambda_k (\xi_{k+1} - \xi_k) = 0, \tag{6}$$

using two exogenous sequences  $\{\varepsilon_k\}$  (summable) and  $\{\lambda_k\}$  (bounded above) of positive numbers. By comparison to other iterative methods, Iusem-Resmerita method has several advantages: it allows very general function  $g$  and very general regularization; at each step, it is theoretically sufficient to compute only an  $\bar{\varepsilon}_k$ -minimizer, thus some error is allowed. On the other hand, Iusem-Resmerita method requires the use of the full norm on  $X$  (and not only a semi-norm), thus the method may be in practice computationally more expensive.



We now apply this general Iusem-Resmerita algorithm [3] to linear ill-posed inverse problems. Large classes of inverse problems can be formulated as operator equations  $Ku = y$ .

We define the residual  $g(u) = S(y, Ku)$  for any  $u \in X$ , where  $S$  is a similarity measure (see, e.g., [12], [8]). The iterative method introduced in [3] can be applied to this exact data case setting and provides weakly\* approximations for the solutions of the equation, provided that at least one solution exists.

Usually, the above equations  $Ku = y$  are ill posed, in the sense that the operator  $K$  may not be continuously invertible which means that small perturbations in the data  $y$  lead to high oscillations in the solutions.

Consider that only noisy data  $y^\delta$  are given, such that

$$S(y^\delta, y) \leq r(\delta), \quad \delta > 0, \tag{7}$$

where  $r = r(\delta)$  is a function of  $\delta$  with

$$\lim_{\delta \rightarrow 0_+} r(\delta) = 0. \tag{8}$$

Denote

$$g^\delta(u) = S(y^\delta, Ku).$$

We show now that the general iterative method presented above yields a regularization method for such problems. We will use the following

**Assumptions (A)**

- The operator  $K : X \rightarrow Y$  is linear and bounded, and yields an ill-posed problem.
- $X$  and  $Y$  are Banach spaces. In addition,  $X$  is the topological dual of a separable Banach space.
- The similarity measure  $S$  is such that
  1. The function  $g^\delta(u) = S(y^\delta, Ku)$  is convex and weakly\* lower semicontinuous.
  - 2.

$$\lim_{\delta \rightarrow 0_+} g^\delta(u_\delta) = 0 \quad \Rightarrow \quad \lim_{\delta \rightarrow 0_+} Ku_\delta = y, \tag{9}$$

whenever  $\{u_\delta\}_{\delta > 0}$  is a net in  $X$ , the last limit being understood with respect to the norm of  $Y$ .

We consider a positive constant parameter  $c$ . The method reads as follows:

**Algorithm 1.** Take  $u_0 \in \text{dom } g^\delta$  and  $\xi_0 \in J_{\varepsilon_0}(u_0)$ .

**Iterative step**

Let  $k \in \mathbb{N}$ . Assume that  $u_k \in \text{dom } g^\delta$  and  $\xi_k \in J_{\varepsilon_k}(u_k)$  are given. We proceed to define  $u_{k+1}$ ,  $\xi_{k+1}$ . Define  $D^{\varepsilon_k}(u, u_k) = h(u) - h(u_k) - \langle \xi_k, u - u_k \rangle + \varepsilon_k$  and  $\bar{\varepsilon}_k = c\varepsilon_{k+1}$ .

Determine  $u_{k+1} \in \text{dom } g^\delta$  as an  $\bar{\varepsilon}_k$ -minimizer of the function  $g_k^\delta(u)$  defined as

$$g_k^\delta(u) = g^\delta(u) + cD^{\varepsilon_k}(u, u_k),$$

that is to say,

$$g^\delta(u_{k+1}) + cD^{\varepsilon_k}(u_{k+1}, u_k) \leq g^\delta(u) + cD^{\varepsilon_k}(u, u_k) + \bar{\varepsilon}_k$$

for all  $u \in \text{dom } g^\delta$ .

Let  $\eta_{k+1} \in \partial g^\delta(u_{k+1})$  and  $\xi_{k+1} \in J_{\varepsilon_{k+1}}(u_{k+1})$  such that

$$\eta_{k+1} + c(\xi_{k+1} - \xi_k) = 0.$$

**A posteriori strategy.** We choose the stopping index based on a discrepancy type principle, similarly to the one in [11]:

$$k_* = \max\{k \in \mathbb{N} : g^\delta(u_k) \geq \tau r(\delta)\}, \tag{10}$$

for some  $\tau > 1$ .

We show that the stopping index is finite and that Algorithm 1 together with the stopping rule stably approximate solutions of the equation (proof included in a longer version of this work [13]).

**Proposition 1.** *Let  $\tilde{u} \in X$  verify  $K\tilde{u} = y$ , assume that inequality (7) is satisfied, assumptions (A) hold and that the sequence  $\{\varepsilon_k\}$  is such that*

$$\sum_{k=1}^{\infty} k\varepsilon_k < \infty. \tag{11}$$

Moreover, let the stopping index  $k_*$  be chosen according to (10). Then  $k_*$  is finite, the sequence  $\{\|u_{k_*}\|\}_\delta$  is bounded and hence, as  $\delta \rightarrow 0$ , there exists a weakly\*-convergent subsequence  $\{u_{k(\delta_n)}\}_n$  in  $X$ . If the following conditions hold, then the limit of each weakly\* convergent subsequence is a solution of  $Ku = y$ :

- i)  $\{k_*\}_{\delta>0}$  is unbounded;
- ii) Weak\*-convergence of  $\{u_{k(\delta_n)}\}_n$  to some  $u \in X$  implies convergence of  $\{Ku_{k(\delta_n)}\}_n$  to  $Ku$ , as  $n \rightarrow \infty$  with respect to the norm topology of  $Y$ .

**A priori strategy.** One could stop Algorithm 1 by using a stopping index which depends on the noise level only, by contrast to the previously chosen  $k_*$  which depends also on the noisy data  $y^\delta$ . More precisely, one chooses

$$k(\delta) \sim \frac{1}{r(\delta)}. \tag{12}$$

One can also show that the sequence  $\{u_{k(\delta)}\}_{\delta>0}$  converges weakly\* to solutions of the equation as  $\delta \rightarrow 0$ .

**Proposition 2.** *Let  $\tilde{u} \in X$  verify  $K\tilde{u} = y$ , assume that inequality (7) is satisfied, assumptions (A) hold and that the sequence  $\{\varepsilon_k\}$  obeys (11). Moreover, let*

the stopping index  $k(\delta)$  be chosen according to [12]. Then the sequence  $\{\|u_{k(\delta)}\|\}_\delta$  is bounded and hence, as  $\delta \rightarrow 0$ , there exists a weakly\*-convergent subsequence  $\{u_{k(\delta_n)}\}_n$  in  $X$ . If the following condition holds, then the limit of each weakly\*-convergent subsequence is a solution of  $Ku = y$ : weak\*-convergence of  $\{u_{k(\delta_n)}\}_n$  to some  $u \in X$  implies convergence of  $\{Ku_{k(\delta_n)}\}_n$  to  $Ku$ , as  $n \rightarrow \infty$  with respect to the norm topology of  $Y$ .

### 3 Several Proximal Point Based Approaches for Image Restoration

We present a few image restoration settings which fit the theoretical framework investigated in the previous section. We assume that noisy blurry data  $f$  corresponding to  $y^\delta$  is given, defined on an open and bounded domain  $\Omega$  of  $\mathbb{R}^N$ . First, we briefly mention prior relevant work in image processing.

In Tadmor et al [4], an iterative procedure for computing hierarchical  $(BV, L^2)$  decompositions has been proposed for image denoising, and this was extended to more general cases for image restoration and segmentation in [5]. For image deblurring in the presence of Gaussian noise, assuming the degradation model  $f = Ku + n$ , the iterative method from [5] computes a sequence  $u_k$ , such that each  $u_{k+1}$  is the minimizer of  $\lambda_0 2^k \|v_k - Ku_{k+1}\|_2^2 + \int_\Omega |Du_{k+1}|$ , where  $v_{-1} = f$ ,  $k = 0, 1, \dots$  and  $v_k = Ku_{k+1} + v_{k+1}$ . The partial sum  $\sum_{j=0}^k u_j$  is a denoised-deblurred version of  $f$ , and converges to  $f$  as  $k \rightarrow \infty$ .

Osher et al [1] proposed an iterative algorithm with quadratic fidelity term  $S$  and a convex regularizing functional  $h$  (e.g. TV-regularizer  $h(u) = \int_\Omega |Du|$ ): starting with  $u_0$ ,  $u_{k+1}$  is a minimizer of the functional

$$g_k(u) = S(f, Ku) + D(u, u_k) = \frac{\lambda}{2} \|f - Ku\|_2^2 + [h(u) - h(u_k) - \langle p_k, u - u_k \rangle], \quad (13)$$

where  $p_k = p_{k-1} + \lambda K^*(f - Ku_k) \in \partial h(u_k)$  and  $\lambda > 0$  is a parameter. The authors of [1] proved the well-definedness and the convergence of iterates  $u_k$ , and presented some applications to denoising or deblurring in the presence of Gaussian noise, obtaining significant improvement over the standard Rudin et al. model [6,14], which is

$$\min_u \left\{ \frac{\lambda}{2} \|f - Ku\|_2^2 + \int_\Omega |Du| \right\}. \quad (14)$$

We also refer to [15] where convergence rates for the iterative method (13) are established.

He et al [2] modified the above iterative algorithm [1] by using the varying parameter  $\frac{1}{2^k \lambda}$  with  $\lambda > 0$  instead of fixed parameter  $\lambda > 0$ , inspired by [16] and [4]:

$$g_k(u) = S(f, u) + D(u, u_k) = S(f, u) + \frac{1}{2^k \lambda} [h(u) - h(u_k) - \langle p_k, u - u_k \rangle], \quad (15)$$

where  $S(f, u) = s(f - u)$  with  $s$  being a nonnegative, convex, and positively homogeneous functional, which is continuous with respect to weak\* convergence in  $BV$ , e.g.  $s(f - u) = \|f - u\|_2^2$  or  $s(f - u) = \|f - u\|_1$ . Thus, the authors proved the well-definedness and the convergence of the algorithm with  $L^1$  fidelity term, which is also the iterative version of the  $L^1$ -TV model considered by Chan and Esedoglu [7], and presented denoising results in the presence of Gaussian noise.

Below we set the general iterative algorithm for image deblurring in the presence of Gaussian and Laplacian noise, with the corresponding (convex) fidelity terms.

### 3.1 Image Deblurring in the Presence of Noise

Let  $X, Y$  be Banach spaces,  $X \subset Y$ , where  $X$  is the dual of a separable Banach space. We consider the standard deblurring-denoising model given by  $f = Ku + n$  where  $f \in Y$  is the observed noisy data,  $K : Y \rightarrow Y$  is a convolution operator with blurring kernel  $K$  (i.e.  $Ku := K * u$ ),  $u \in X$  is the ideal image we want to recover, and  $n$  is noise.

Here, we present two noise models in infinite dimension prompted by the corresponding finite dimensional models based on the conditional probability  $p(f|Ku)$ : the Gaussian model and the Laplace model. In finite dimensional spaces, the conditional probability  $p(f|Ku)$  of the data  $f$  with given image  $Ku$  is the component of the Bayesian model that is influenced by the type of distribution of the noise (and hence the noisy data  $f$ ).

Assuming  $X = BV(\Omega)$  and  $Y = L^p(\Omega)$  with  $p = 1$  or  $2$ , we have

$$h(u) = \frac{1}{2} \|u\|_{BV}^2 = \frac{1}{2} \left( \int_{\Omega} |u| dx + \int_{\Omega} |Du| \right)^2.$$

Here  $\Omega$  is a bounded and open subset of  $\mathbb{R}^N$ .

In addition, we consider convex functions of the form  $g(u) = S(f, Ku)$  for any  $u \in X$ , where  $S$  is convex with respect to  $u$  for a fixed  $f$ . Then, we propose the following general iterative algorithm to recover  $u$ :

**Algorithm 4.1.** Let  $u_0 = 0, \xi_0 = 0, \varepsilon_0 = 0$  and iterate for  $k \in \mathbb{Z}, k \geq 0$ .

- Given  $(u_k, \xi_k)$ , define  $\bar{\varepsilon}_k = c\varepsilon_{k+1}$ , and compute  $u_{k+1}$  as a  $\bar{\varepsilon}_k$ -minimizer of the functional  $g_k(u) = S(f, Ku) + c[h(u) - h(u_k) - \langle \xi_k, u - u_k \rangle + \varepsilon_k]$ .
- Determine  $\eta_{k+1} \in \partial_u S(f, Ku_{k+1})$  and  $\xi_{k+1} \in J_{\varepsilon_{k+1}}(u_{k+1})$  such that  $\eta_{k+1} + c(\xi_{k+1} - \xi_k) = 0$ .

Note that we use the gradient descent method to minimize  $g_k(u)$ . In what follows, in practice, we assume that we work with functions  $u \in W^{1,1}(\Omega) \subset BV(\Omega)$ . Also, we make the functional  $h(u)$  differentiable by substituting it with  $h(u) \approx \frac{1}{2} \left( \int_{\Omega} \sqrt{\varepsilon^2 + u^2} dx + \int_{\Omega} \sqrt{\varepsilon^2 + |\nabla u|^2} dx \right)^2$ ,  $\varepsilon > 0$  small. The subgradient in this case becomes

$$\partial h(u) \approx \left( \int_{\Omega} \sqrt{\varepsilon^2 + u^2} + \sqrt{\varepsilon^2 + |\nabla u|^2} dx \right) \left[ \frac{u}{\sqrt{\varepsilon^2 + u^2}} - \nabla \cdot \frac{\nabla u}{\sqrt{\varepsilon^2 + u^2}} \right].$$

Also, we refer to [17, Section 3.4.1] for the relation between Gateaux differentiability and  $\bar{\varepsilon}_k$ -minimizers. If  $u$  is an  $\bar{\varepsilon}_k$ -minimizer of the Gateaux-differentiable function  $g_k(u)$ , then we must have  $\|\partial g_k(u)\| \leq \bar{\varepsilon}_k$ . In practice, we use time-dependent gradient descent to approximate an  $\bar{\varepsilon}_k$ -minimizer  $u$  by solving  $\frac{\partial u}{\partial t} = -\partial g_k(u) + \bar{\varepsilon}_k$  to steady state.

*Remark 1.* We can start with  $u_0 = 0, \xi_0 = 0, \varepsilon_0 = 0$ . Although our theory considers positive parameters  $\varepsilon_k$  in order to ensure existence of the iterates  $u_k$ , one could still initialize the algorithm with  $u_0 = 0, \xi_0 = 0, \varepsilon_0 = 0$  in many situations, including the particular ones investigated below. In such cases, existence of  $u_1$  and  $\xi_1$  is not based on the surjectivity result employed in [3], but rather on direct analysis of the function  $S(f, Ku) + ch(u)$  to be minimized.

**Gaussian noise.** If the data is  $f = Ku + n \in Y = L^2(\Omega)$  with Gaussian distributed noise and with the expectation  $Ku$ , the conditional probability  $p(f|Ku)$  is described as  $p(f|Ku) \sim e^{-\frac{\|f-Ku\|_2^2}{2\sigma^2}}$ , where  $\sigma^2$  is the variance of the noise  $n$ . Maximizing  $p(f|Ku)$  with respect to  $u$ , is equivalent to minimizing  $-\ln p(f|Ku)$ , thus we obtain a convex fidelity term to be minimized for  $u \in BV(\Omega)$ ,  $S(f, Ku) = \frac{1}{2}\|f - Ku\|_2^2$ . The function  $g(u) = S(f, Ku)$  satisfies the conditions enforced in Assumptions (A) in dimension one and two. Moreover, let  $r(\delta) = \delta^2/2$  (see [7]).

Since such a quadratic  $S$  is Gateaux-differentiable, its subgradient is given by  $\partial_u S(f, Ku) = K^*(Ku - f)$  which leads to  $\xi_{k+1} = \xi_k - \frac{1}{c}K^*(Ku_{k+1} - f)$ . We propose the following numerical algorithm:

**Numerical Algorithm.**

- I.** Let  $u_0 = 0, \xi_0 = 0, \varepsilon_0 = 0$  and iterate for  $k \in \mathbb{Z}, k \geq 0$  until  $\|f - Ku_{k+1}\|_2 \leq \sigma$ :
- For  $u = u_{k+1}$ , use  $\frac{\partial u}{\partial t} = K^*(f - Ku) - c[\partial h(u) - \xi_k] + c\varepsilon_{k+1}$
  - For  $\xi_{k+1}$ , use  $\xi_{k+1} = \xi_k + \frac{1}{c}K^*(f - Ku_{k+1})$ .

In addition, following [1], we let  $\xi_k = \frac{K^*v_k}{c}$  so that we have  $v_{k+1} = v_k + (f - Ku_{k+1})$ .

With  $v_0 = 0$ , since  $c\xi_0 = 0 = K^*0 = K^*v_0$ , we may conclude inductively that  $c\xi_k \in \text{Range}(K^*)$ , and hence there exists  $v_k \in Y^* = L^2(\Omega)$  such that  $c\xi_k = K^*v_k$ . Hence, we can have the following alternative numerical algorithm:

- II.** Let  $u_0 = 0, v_0 = 0, \varepsilon_0 = 0$  and iterate for  $k \in \mathbb{Z}, k \geq 0$  until  $\|f - Ku_{k+1}\|_2 \leq \sigma$ :
- For  $u = u_{k+1}$ , use  $\frac{\partial u}{\partial t} = K^*(f + v_k - Ku) - c\partial h(u) + c\varepsilon_{k+1}$
  - For  $v_{k+1}$ , use  $v_{k+1} = v_k + (f - Ku_{k+1})$ .

**Laplacian noise.** If the data is  $f = Ku + n \in Y = L^1(\Omega)$  with  $n$  being a Laplacian distributed random variable with mean zero and variance  $2\sigma^2$ , we

**Table 1.** RESULTS USING DIFFERENT  $\varepsilon_k$  ( $u_*$ : ORIGINAL IMAGE)

GAUSSIAN NOISE, SHAPE IMAGE,  $\sigma = \|f - K * u_*\|_2 = 15$  (FIG. 11)

$\varepsilon_k$ ( $k > 0$ )		$a = \ f - K * u_k\ _2$	$b = \text{RMSE vs } k = 1, 2, 3, 4, 5$	$\lambda = 0.1$		
0	a	28.8376	16.1033	<b>14.9578</b>	14.3887	13.9292
	b	38.3389	27.7436	<b>22.0875</b>	20.1510	19.4987
$\frac{1}{2^k}$	a	28.8054	16.1075	<b>14.9587</b>	14.3877	13.9277
	b	38.3168	27.7422	<b>22.0875</b>	20.1473	19.4980

LAPLACIAN NOISE, RECTANGLES IMAGE,  $\sigma = \|f - K * u_*\|_1 = 10$  (FIG. 5)

$\varepsilon_k$ ( $k > 0$ )		$a = \ f - K * u_k\ _1$	$b = \text{RMSE vs } k = 1, 2, 3, 4, 5$	$\lambda = 0.05$		
0	a	16.2208	10.3479	<b>9.9939</b>	9.9768	9.9615
	b	25.2332	6.1003	<b>2.4687</b>	2.2553	2.4489
$\frac{1}{2^k}$	a	15.8850	10.3339	<b>9.9935</b>	9.9768	9.9617
	b	24.3540	6.0411	<b>2.4938</b>	2.2898	2.4733

**Table 2.** STOPPING CRITERIA AND COMPARISONS,  $\sigma^2 \sim$  NOISE VARIANCE

Noise Model	Stopping criteria	Comparison with
Gaussian	$\ f - K * u_k\ _2 \leq \sigma$	iterative algorithm using TV (13) or RO (14)
Laplacian	$\ f - K * u_k\ _1 \leq \sigma$	iterative (15) or one-step $L^1$ -TV deblurring model

have  $p(f|Ku) \sim e^{-\frac{\|f-Ku\|_1}{\sigma}}$ . Then, similarly, we minimize with respect to  $u$  the quantity  $-\ln p(f|Ku)$ , thus we are led to consider the convex fidelity term

$$S(f, Ku) = \int_{\Omega} |f - Ku| dx.$$

Moreover, let  $r(\delta) = \delta$ . Again, the function  $g(u) = S(f, Ku)$  satisfies the conditions in Assumptions (A) in dimension one and two.

Unless  $Ku \equiv f$ , one can think of  $\partial_u S(f, Ku) = K^* \text{sign}(Ku - f)$  almost everywhere, and moreover we have

$$\xi_{k+1} = \xi_k - \frac{1}{c} K^* \text{sign}(Ku_{k+1} - f) \quad a.e.$$

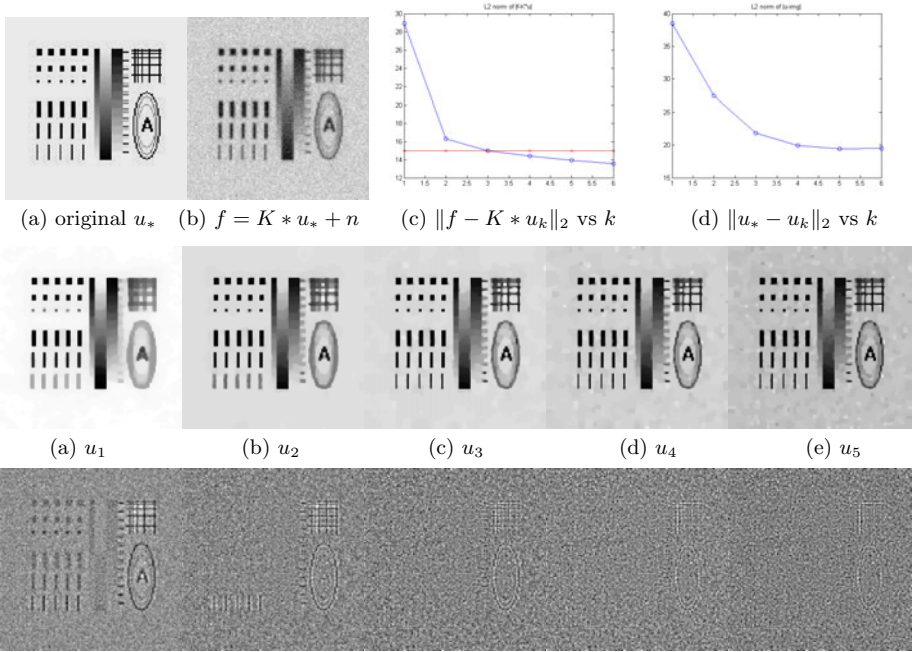
We propose the following numerical algorithm:

**Numerical algorithm.**

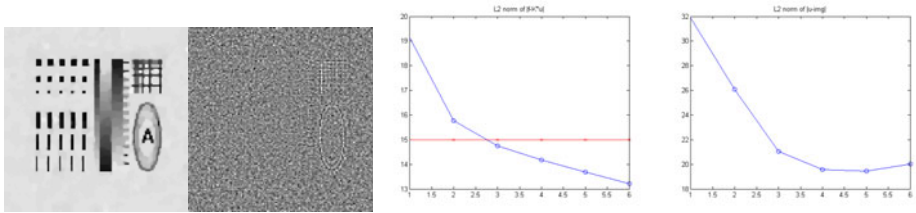
I. Let  $u_0 = 0, \xi_0 = 0, \varepsilon_0 = 0$  and iterate for  $k \in \mathbb{Z}, k \geq 0$  until  $\|f - Ku_{k+1}\|_1 \leq \sigma$ :

- For  $u = u_{k+1}$ , use  $\frac{\partial u}{\partial t} = K^* \text{sign}(f - Ku) - c[\partial h(u) - \xi_k] + c\varepsilon_{k+1}$
- For  $\xi_{k+1}$ , use  $\xi_{k+1} = \xi_k + \frac{1}{c} K^* \text{sign}(f - Ku_{k+1})$ .

Now again letting  $\xi_k = \frac{K^* v_k}{c}$ , we can have  $v_{k+1} = v_k + \text{sign}(f - Ku_{k+1}) \quad a.e.$



**Fig. 1.** Results for the Gaussian noise model obtained by the proposed iterative method. 2nd and 3rd row: recovered images  $u_k$  and the corresponding residuals  $f - K * u_k$ . Data: Gaussian blur kernel  $K$  with the standard deviation  $\sigma_b = 0.7$ , and Gaussian noise with  $\sigma_n = 15$ ,  $\lambda = 0.1$ .  $\|f - K * u_3\|_2 = 14.9658$ .  $u_3$  is the best recovered image (RMSE=21.8608).

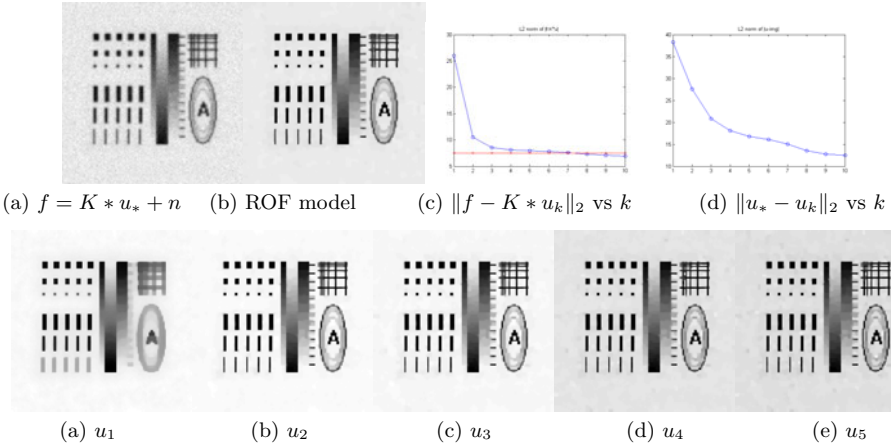


**Fig. 2.** Results of the iterative algorithm (13) proposed by Osher et al with the same data in Fig. 1. The best recovered image  $u_3$  ( $\|f - K * u_3\|_2 = 14.7594$ , RMSE=21.0500), residual  $f - K * u_3$ , and energies  $\|f - K * u_k\|_2$ ,  $\|u_* - u_k\|_2$  vs  $k$ .

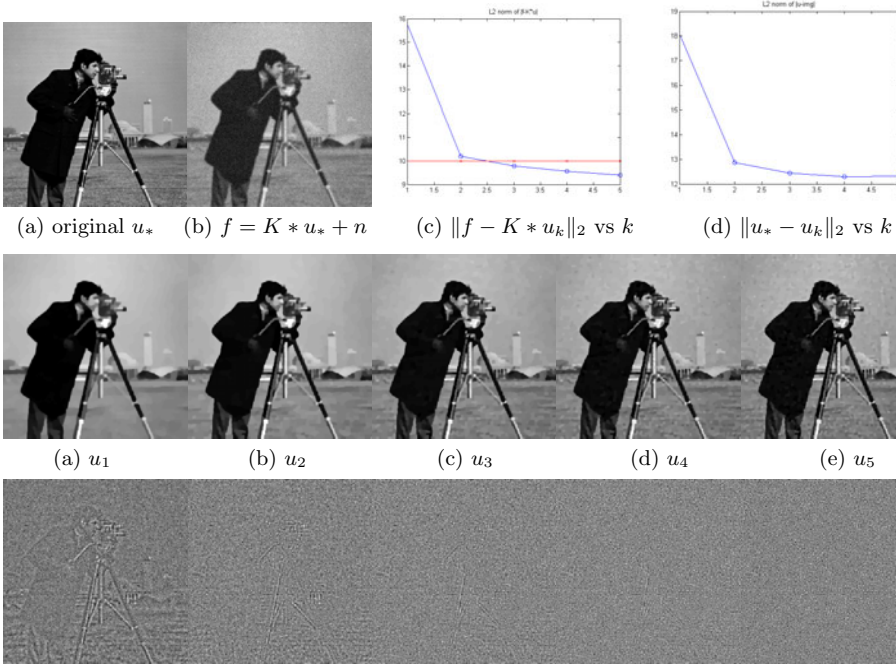
With  $v_0 = 0$ , since  $c\xi_0 = 0 = K*0 = K*v_0$ , we may conclude inductively that  $c\xi_k \in Range(K^*)$ , and hence there exists  $v_k \in Y^* = L^\infty(\Omega)$  such that  $c\xi_k = K*v_k$ . Hence, we have the alternative numerical algorithm:

**II.** Let  $u_0 = 0$ ,  $v_0 = 0$ ,  $\varepsilon_0 = 0$  and iterate for  $k \in \mathbb{Z}$ ,  $k \geq 0$  until  $\|f - K u_{k+1}\|_1 \leq \sigma$ :

- For  $u = u_{k+1}$ , use  $\frac{\partial u}{\partial t} = K^*[sign(f - K u) + v_k] - c\partial h(u) + c\varepsilon_{k+1}$
- For  $v_{k+1}$ , use  $v_{k+1} = v_k + sign(f - K u_{k+1})$ .

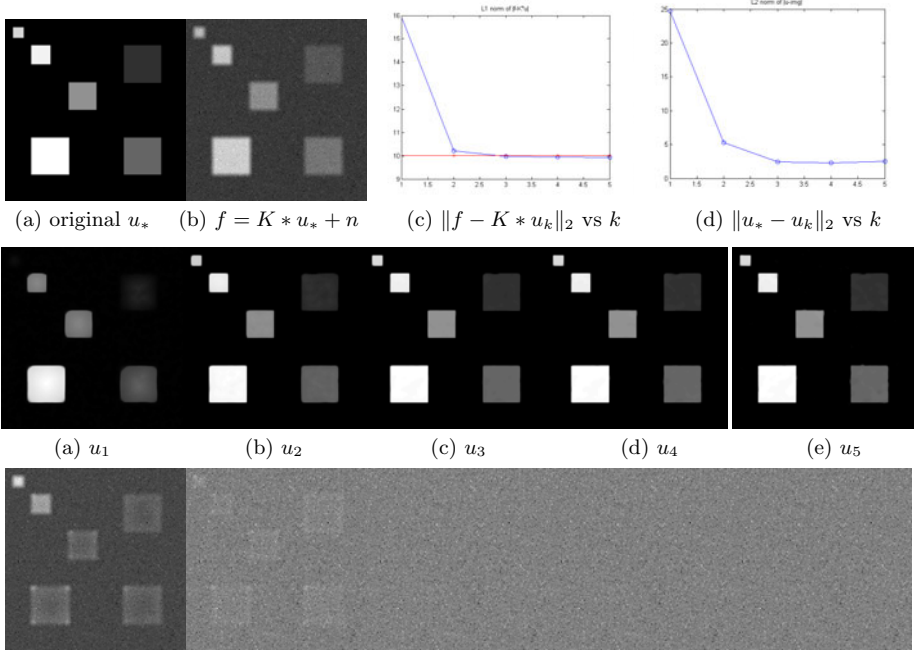


**Fig. 3.** Stopping index  $k(\delta) \sim \delta^{-1}$  and comparison with RO model (RMSE=16.5007). Data: same blur kernel  $K$  and parameter  $c = 0.1$  with Fig. 1 but different Gaussian noise with  $\sigma_n = 7.5$ .  $u_8$  is the best recovered image (RMSE=13.5407).



**Fig. 4.** Results for the Gaussian noise model obtained by the proposed method. Data: Gaussian blur kernel  $K$  with  $\sigma_b = 1$ , and Gaussian noise with  $\sigma_n = 10$ . Parameters:  $c = 0.1$ .  $u_3$  is the best recovered image (RMSE=12.2217).



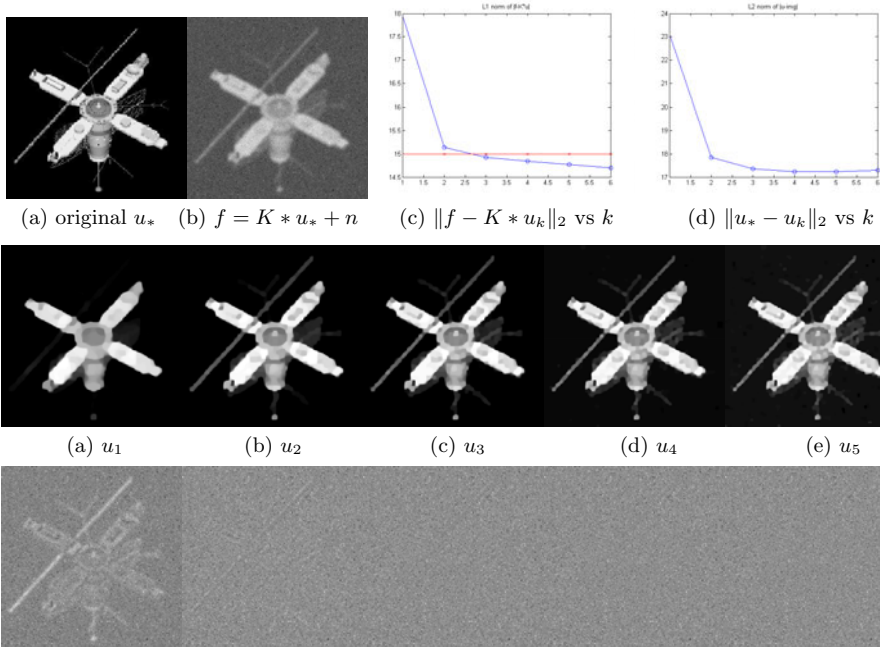


**Fig. 5.** Results for the Laplacian noise model obtained by the proposed method. Data: Gaussian blur kernel  $K$  with  $\sigma_b = 3$ , and Laplacian noise with  $\sigma_n = 10$ . Parameters:  $\lambda = 0.05$ .  $\|f - K * u_3\|_1 = 9.9629$ .  $u_3$  is the best recovered image (RMSE=2.4417).

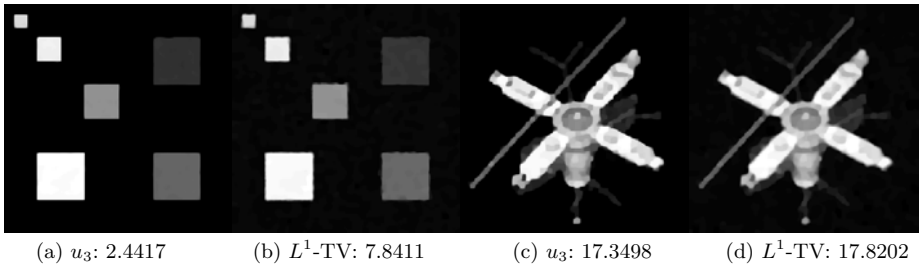
### 4 Numerical Results

We assume  $|\Omega| = 1$ . First, with fix the parameter  $\lambda$ , and then we test each model with different  $\epsilon_k$ , either  $\epsilon_k = \frac{1}{2k}$  or  $\epsilon_k = 0$ . These different values of  $\epsilon_k$  produce almost the same results according to the measured values in Table 1 as well as visually. Thus, in all the other examples, we numerically set  $\epsilon_k = 0$ . Since our experimental results are done on artificial tests, we can compare the restored images  $u_k$  with the true image  $u_*$ .

First, we consider the residual  $S(f, Ku_k)$  and the  $L^2$  distance between iterates  $u_k$  and the original image  $u_*$ ,  $\|u_* - u_k\|_2$  (or root mean square error, denoted RMSE). As  $k$  increases, the image  $u_k$  recovers more details and fine scales, and eventually gets noise back. Thus, in practice, the residual  $g(u_k) = S(f, Ku_k)$  keeps decreasing even when  $\epsilon_k \neq 0$  (see Table 1), while  $\|u_* - u_k\|_2$  has a minimum value at some  $k'$ . But, note that  $k'$  does not correspond to the optimal  $k_* = \min\{k : g(u_k) = S(f, Ku_k) \leq \sigma^2\}$ , i.e  $k' > k_*$ , which is not surprising because in the presence of blur and noise,  $u_{k'}$  can have lower RMSE since  $u_{k'}$  may become sharper than  $u_{k_*}$  even though  $u_{k'}$  becomes noisier than  $u_{k_*}$ . However, the visual quality is also best at the optimal  $k_*$ . For example, in Fig. 5 with Gaussian noise,



**Fig. 6.** Results for the Laplacian noise model obtained by the proposed method. Data: Gaussian blur kernel  $K$  with  $\sigma_b = 2$ , and Laplacian noise with  $\sigma_n = 15$ . Parameters:  $\lambda = 0.02$ .  $\|f - K * u_3\|_1 = 14.9234$ .  $u_3$  is the best recovered image (RMSE=17.3498).



**Fig. 7.** Comparison with one-step  $L^1$ -TV [7]. (a), (c): our iterative method. (b), (d): one-step  $L^1$ -TV ( $\|f - K * u\|_1$ : (b) 9.8649 , (d) 14.9650 ). Recovered images  $u$  and RMSE values.

$u_3$  ( $k_* = 3$ ) recovers the details well enough leading to the best visual quality while  $\|u_* - u_k\|_2$  is still decreasing, and  $u_k$  for  $k > 3$  becomes noisier. Thus the optimal  $k_*$  is a reasonable choice for Gaussian and Laplace noise models.

In Figures 1-4, we test the Gaussian noise model using  $L^2$  fidelity term, and moreover we compare our result with the iterative algorithm (13) proposed by Osher et al. In Figures 1 and 4,  $u_3$  recovers texture parts or details better than in the previous iterates, with less noise, while the next iterate  $u_4$  becomes noisier.

In addition, from Figures 1 and 2 we observe that our iterative algorithm and the one from (13) proposed by Osher et al provide similar best recovered images and similar behavior. Fig. 3 verifies the a-priori property for the stopping index (12); with less noise ( $\sigma_n = 7.5$ ), the stopping index  $k_* = 8$  is twice larger than the one ( $k_* = 3$ ) with  $\sigma_n = 15$ . Moreover, Fig. 3 shows that our iterative scheme provides superior result to the Rudin-Osher model [?] by recovering details or texture parts better, leading to much better RMSE.

In Figures 5-7, we show the recovered images  $u_k$  in the presence of Laplacian noise with  $L^1$  fidelity term, and we compare our results with the iterative algorithm (15) proposed by He et al and the one-step  $L^1$ -TV model (analyzed by Chan and Esedoglu [7] when  $K = I$ ). In Figures 5 and 6,  $u_k$  restores fine scales and becomes sharper until the optimal  $k_* = 3, 2$  respectively, and  $u_{k_*}$  gives cleaner images than  $u_k$  for  $k > k_*$ . We have also compared with the iterative algorithm (15), which produces slightly worse result than ours. In Fig. 7 we observe that our iterative method gives cleaner and sharper images and moreover smaller RMSE than by the one-step  $L^1$ -TV model with blur.

## 5 Conclusion

We introduced a generalized iterative regularization method based on the norm square for image restoration models with general convex fidelity terms. We applied the proximal point method [3] using inexact Bregman distance to several ill-posed problems in image processing (image deblurring in the presence of noise). The numerical experiments indicate that for deblurring in the presence of noise, the iterative procedure yields high quality reconstructions and superior results to the one-step gradient-descent models and similar with existing iterative models. For an extended version of this work, we refer the reader to [13], where the details of the proofs are given, the data fidelity term arising from Poisson noise distribution is also considered, together with the deblurring problem using cartoon + texture representation for better texture preservation in the restoration problem.

## References

1. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation based image restoration. *Multiscale Modelling and Simulation* 4, 460–489 (2005)
2. He, L., Osher, S., Burger, M.: Iterative total variation regularization with non-quadratic fidelity. *J. Math. Imag. Vision* 26, 167–184 (2005)
3. Iusem, A.N., Resmerita, E.: A proximal point method in nonreflexive banach spaces. *Set-Valued and Variational Analysis* 18, 109–120 (2010)
4. Tadmor, E., Nezzar, S., Vese, L.: A multiscale image representation using hierarchical (BV,  $L^2$ ) decompositions. *Multiscale Model. Simul.* 2, 554–579 (2004)
5. Tadmor, E., Nezzar, S., Vese, L.: Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation. *Commun. Math. Sci.* 6, 281–307 (2008)

6. Rudin, L., Osher, S.J., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* 60, 259–268 (1992)
7. Chan, T.F., Esedoglu, S.: Aspects of total variation regularized l1 function approximation. *SIAM J. Appl. Math.* 65, 1817–1837 (2005)
8. Benning, M., Burger, M.: Error estimates for variational models with non-gaussian noise. *UCLA CAM Report 09-40* (2009)
9. Gossez, J.P.: Opérateurs monotones nonlineaires dans les espaces de banach non-reflexifs. *J. Math. Anal. Appl.* 34, 371–395 (1971)
10. Marques, A.M., Svaiter, B.: On the surjectivity properties of perturbations of maximal monotone operators in non-reflexive banach spaces (to be published)
11. Ekeland, I., Temam, R.: *Convex analysis and variational problems*. SIAM, Philadelphia (1999)
12. Pöschl, C.: Regularization with a similarity measure. PhD Thesis, University of Innsbruck (2008)
13. Jung, M., Resmerita, E., Vese, L.: Dual norm based iterative methods for image restoration. *UCLA C.A.M. Report 09-88* (2009)
14. Rudin, L., Osher, S.: Total variation based image restoration with free local constraints. *IEEE ICIP*, 31–35 (1994)
15. Burger, M., Resmerita, E., He, L.: Error estimation for bregman iterations and inverse scale space methods in image restoration. *Computing* (81)
16. Scherzer, O., Groetsch, C.: Inverse scale space theory for inverse problems. In: Kerckhove, M. (ed.) *Scale-Space 2001*. LNCS, vol. 2106, pp. 317–325. Springer, Heidelberg (2001)
17. Attouch, H., Buttazzo, G., Michaille, G.: *Variational analysis in Sobolev and BV spaces* (2006)

# One-Shot Optimal Exposure Control

David Ilstrup and Roberto Manduchi

University of California, Santa Cruz

**Abstract.** We introduce an algorithm to estimate the optimal exposure parameters from the analysis of a single, possibly under- or over-exposed, image. This algorithm relies on a new quantitative measure of exposure quality, based on the average rendering error, that is, the difference between the original irradiance and its reconstructed value after processing and quantization. In order to estimate the exposure quality in the presence of saturated pixels, we fit a log-normal distribution to the brightness data, computed from the unsaturated pixels. Experimental results are presented comparing the estimated vs. “ground truth” optimal exposure parameters under various illumination conditions.

**Keywords:** Exposure control.

## 1 Introduction

Correct image exposure is critical for virtually any computer vision application. If the image is under- or over-exposed, features or texture are lost, colors are washed out, and the overall perceptual quality of the image is decreased. Correct exposure means that the best possible use is made of the quantization levels provided by the digitization system – in other words, that the *rendering error* due to the non-ideal imaging system is minimized, where the rendering error is the difference between the true irradiance at a pixel and what can be reconstructed based on the measured brightness.

In this paper we propose a quantitative measure for the quality of exposure, along with an algorithm to estimate the optimal exposure based on single, possibly under- or over-exposed, image. By using only one image (rather than several images taken at different exposures) our algorithm enables a fast mechanism for exposure control, a useful characteristic in many contexts. For example, vision system mounted on mobile robots need to adapt quickly to new scenes imaged as the robots moves around. Surveillance systems require prompt response to sudden changes in illumination, such as a light turned on or off. Likewise, through-the-lens (TTL) digital cameras systems for the consumer or professional market may benefit from fast and accurate exposure control.

Our definition of exposure quality requires estimation of the rendering error and of its expected behavior with varying exposure parameters. Unfortunately, the rendering error can only be computed if the original, unprocessed irradiance data is available - a luxury that is not usually available. In particular, if some of the pixels are saturated, their value and thus their rendering error is simply

unknown. We note in passing that, in general, a correctly exposed image contains a certain amount of saturated pixels: an exposure control strategy that simply avoids saturation is usually sub-optimal. We propose a procedure to estimate the rendering error for saturated pixels based on a prior statistical model of the image brightness. Basically, we fit a parametric distribution model to the unsaturated data; the “tail” of this distribution tells us what to expect beyond the saturation point. Computing this model boils down to a problem of parameter estimation from right-censored data, a well-studied statistical technique. Combined with the brightness histogram of the unsaturated data, the model-based distribution for the saturated data allows us to predict how the rendering error changes as one increases or decreases the exposure time, and thus to estimate the optimal exposure, as the one that minimizes the rendering error.

This paper is organized as follows. After presenting related work in Sec. 2, we introduce our quantitative definition of exposure quality in Sec. 3. Next Sec. 4 shows how the exposure quality can be evaluated from a single image, and introduces our parametric statistical model for the unobserved (saturated) pixels. This concept is brought forward in Sec. 5, where we describe how to estimate the rendering error for various exposures from observation of an image at a fixed exposure, enabling a mechanism for estimating the optimal exposure. Quantitative experiments are described in Sec. 6.

## 2 Related Work

Much of the existing literature for automatic exposure control appears as patents (e.g. [1,2,3]). A common theme in all these works is the use of some *scene evaluation* heuristics. Scene evaluation can range from relatively simple goals such identifying back-lit and front-lit scenes [4] to the complex task of face detection [5]. Once the most important areas of the scene are determined, exposure control is adjusted so that some statistic of these pixels, such as the mean, reaches a desired value, often near the middle of the pixel range (e.g. 128 for an 8-bit image). Adjustment is normally achieved via dynamic control algorithms [6,7,8].

A per-pixel control algorithm where the objective function is based on a model of the camera’s response function is given in [9]. The goal of this system is to modify the exposure of each pixel (or, in this case, the transmittance of a coupled spatial light modulator) so that the irradiant energy is just below saturation. If the pixel is unsaturated, then the next exposure is computed trivially. If the pixel is saturated, then the exposure is decreased by a large constant fraction.

Schulz *et al.* [10] measure the goodness of exposure by the integral of the brightness histogram within the bounds of the camera’s dynamic range (from the minimum brightness above noise level to the maximum brightness before saturation). Although this measure of goodness may resemble the one proposed in this paper, it lacks a sound theoretical justification, and may give very different results from ours.

Recent work on high-dynamic range (HDR) imaging has addressed the issue of how to efficiently combine low-dynamic range (LDR) images into an HDR stack

(see *e.g.* [11]). The goal is to find a *minimal image-bracketing* set that covers all of the scene dynamics. In order to minimize the acquisition time, one needs an efficient strategy for selecting the exposure of the next LDR image to take. Barakat *et al.* [12] propose three different approaches: Blind acquisition; Clairvoyant acquisition; and Blind acquisition with feedback. Under this terminology, our proposed approach can be defined as a blind acquisition system that tries to best capture the scene dynamics after observation of just one previous image.

### 3 Exposure Quality: A Quantitative Definition

A pixel in a camera’s focal plane converts irradiant energy into a number (*brightness*). For a given *exposure time* (or, concisely, *exposure*)  $T$ , the irradiant energy  $I_T$  is a function of the irradiant power integrated over the pixel’s surface<sup>1</sup>,  $I$ :

$$I_T = I \cdot T \quad (1)$$

Note that the irradiant power  $I$  is approximately a linear function of the iris aperture area, especially for pixels near the center of the image, which adds one multiplicative factor in [11]. We will assume constant iris aperture in this paper.

Conversion from irradiant energy  $I_T$  to brightness  $B_T$  normally comprises two steps: (a) transformation of  $I_T$  into electrical charge; (b) quantization of a voltage signal that is a function of this charge. For the sake of simplicity, subsequent operations on the digital data (such as white balancing, gamma correction, or sub-quantization) are neglected in this work. We note that, at least for cameras in the higher market segments, these operations can usually be overridden by proper configuration setting.

Formally, this conversion can be represented as follows:

$$B_T = Q(f(I_T)) \quad (2)$$

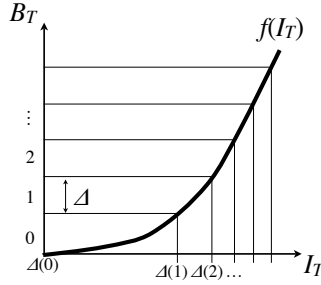
The *sensor’s characteristic*  $f$  can usually be modeled as an invertible noisy function, and can be estimated using standard methods (see *e.g.* [13,14]). The inverse function of  $f$  will be denoted by  $g$ :  $g(f(I_T)) = I_T$ . Note that embedded in the sensor’s characteristic  $f$  is also the variable gain amplification, which can be also used as an exposure parameter.

The non-invertible quantization operator  $Q$  maps values of  $f(I_T)$  into numbers between 0 and  $2^N - 1$ , where  $N$  is the number of bits. Using a mid-tread model [15], the quantization operation can be formalized as follows:

$$Q(x) = \begin{cases} \text{round}(x/\Delta) & , x < (2^N - 1)\Delta \\ 2^N - 1 & , x \geq (2^N - 1)\Delta \end{cases} \quad (3)$$

where  $\Delta$  is the quantization step. In practice, values of  $I_T$  within an *equivalent bin*  $[\Delta(i), \Delta(i + 1)]$ , where  $\Delta(i) = g(i\Delta)$ , are mapped to  $B_T = i$  (see Fig. [1]).

<sup>1</sup> Without loss of generality, it will be assumed that the pixel has unit area in an appropriate scale.



**Fig. 1.** Conversion of irradiant energy  $I_T$  into brightness  $B_T$

Values of  $I_T$  above  $g((2^N - 1)\Delta)$  are saturated to  $2^N - 1$ . Note that in the case of linear sensor characteristic ( $f(x) = ax$ ), increasing the exposure by a factor of  $k$  ( $T \rightarrow kT$ ) is completely equivalent to reducing the quantization step by the same factor ( $\Delta \rightarrow \Delta/k$ ).

We define by *rendering error*  $e_T$  at a pixel the difference between the true irradiant power,  $I$ , and the best reconstruction from the brightness  $B_T$ :

$$e_T = I - g(B_T \Delta) / T \tag{4}$$

The irradiant power  $I$  is independent of the exposure setting (for constant iris aperture) and thus represents a more natural domain for the definition of rendering error  $e_T$  than the radiant energy  $I_T$ . Note that the dependence of  $e_T$  on  $T$  as we analyze it is only due to the presence of the quantizer (but see Appendix B). When  $I_T < g((2^N - 1)\Delta)$ , the signal is said to be in the *granular region*.

If the equivalent quantization bins are small enough that the sensor’s characteristic  $f(I_T)$  has constant slope within each individual bin, then one easily sees that, when  $I_T$  is within the  $i$ -th equivalent bin, the error  $e_T$  is confined between  $-\alpha(i)\Delta/2$  and  $\alpha(i)\Delta/2$ , where  $\alpha(i) = g'((i + 1/2)\Delta)$ . When  $I_T > g((2^N - 1)\Delta)$ , the signal is said to be in the *overload region*, generating an unbounded error (meaning that the pixel is saturated).

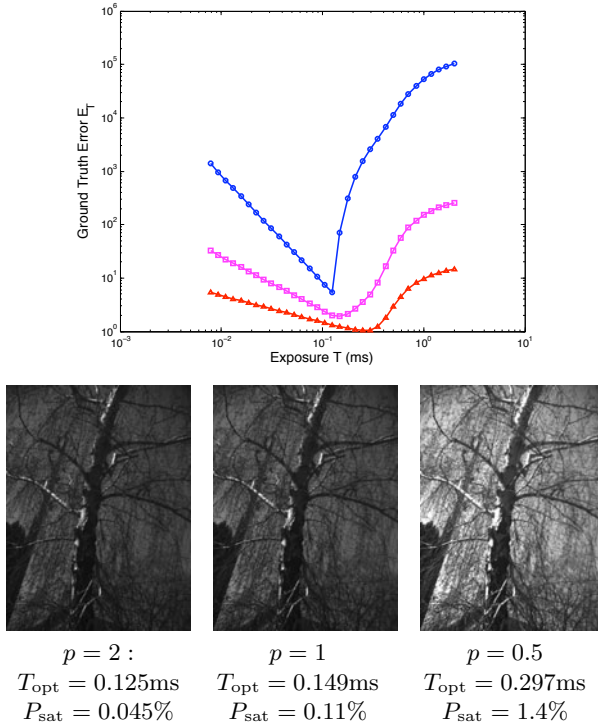
In order to assess the effect of quantization, we can define a positive measure of the rendering error  $L(e_T(m))$  at each pixel  $m$ , and average it over the whole image:

$$E_T = \sum_{m=1}^N L(e_T(m)) / M \tag{5}$$

where  $M$  is number of pixels in the image. The *optimal exposure* for a particular scene is the value of the exposure  $T$  that minimizes the associated error  $E_T$ . The goal of exposure control is thus one of finding the optimal exposure, given the observations (images) available. In this paper, we describe an algorithm that attempts to find the optimal exposure from analysis of a single image, taken with a known (and presumably suboptimal) exposure  $T_0$ .

Our definition of exposure quality promotes a “good” balance between the overload error due to saturation and the granular error for unsaturated pixels. The optimal exposure depends on the chosen error measure  $L$ . One may choose,





**Fig. 2.** The error  $E_T$  as a function of exposure  $T$  for an 8-bit system with  $L(e_T) = |e_T|^p$ . Blue circles:  $p = 2$ . Magenta squares:  $p = 1$ . Red triangles:  $p = 0.5$ . The minimizer of each curve represents the optimal exposure  $T_{\text{opt}}$  for the corresponding measure. The optimally exposed image for the each measures are also shown, along with the percentage of saturated pixels  $P_{\text{sat}}$ .

for example,  $L(e_T) = |e_T|^p$  for an appropriate value of the exponent  $p$ . Larger values of  $p$  penalize the overload error more (since it can grow unbounded). For example, in Fig. 2 we show the error  $E_T$  for  $p = 0.5, 1$  and  $2$  as a function of  $T$  using 8-bit pixel depth for a particular scene. (For this and other experiments we synthesized 8-bit images from a 12-bit image as discussed in Appendix A, and used data from the 12-bit image as “ground truth”). Optimally exposed images for the three measures chosen (corresponding to the minimizers of the curves) are also shown in the image. Note that using  $p = 0.5$ , a brighter image with more saturated pixels (1.4% of the image) is obtained, while  $p = 2$  allows for much fewer saturated pixels only. Other error measures (*e.g.* robust measures such as Tukey’s biweight function) could also be considered. For all experiments in this paper, we used the measure  $L(e_T) = |e_T|$ .

## 4 Evaluating Exposure Quality

Computation of (5) is only feasible if the irradiance  $I$  is known for each pixel, an unrealistic assumption in any practical situation. Instead, one may estimate

$E_T$  by means of the expected error measure over a suitable probability density. More precisely, we model the values of irradiant power at the pixels as samples independently drawn from a density  $p(I)$ . Thus, the expected value  $E_T$  of  $L(e_T)$  can be written as:

$$E_T = \int_0^\infty L(e_T)p(I) dI = E_T^g + E_T^o \quad (6)$$

$$E_T^g = \sum_{i=0}^{2^N-2} \int_{\Delta(i)/T}^{\Delta(i+1)/T} L(e_T)p(I) dI ; E_T^o = \int_{g((2^N-1)\Delta)/T}^\infty L(e_T)p(I) dI \quad (7)$$

In the following analysis we only consider the effect of quantization noise. While the overall level and variance of photon noise can be significant, in Appendix B we argue that this has little effect on the optimum exposure value  $T_{\text{opt}}$ , especially compared to the effect of changing  $L$  in (5) or changes in the distribution of irradiance at the sensor.

If the density  $p(I)$  can be considered constant within each equivalent bin (“high rate” assumption [15]), and still assuming that the sensor characteristic has linear slope within each equivalent bin, the granular error is uniformly distributed within  $-\alpha(i)\Delta/2T$  and  $\alpha(i)\Delta/2T$ . This enables easy computation of the granular error  $E_T^g$ . The dependence of  $E_T^g$  on  $T$  is normally complex, except when the sensor has a linear characteristic  $f(I_T)$ , in which case the following identity holds:

$$E_T^g = \Phi_T \cdot \text{Prob}(I < (2^N - 1)\Delta/T) \quad (8)$$

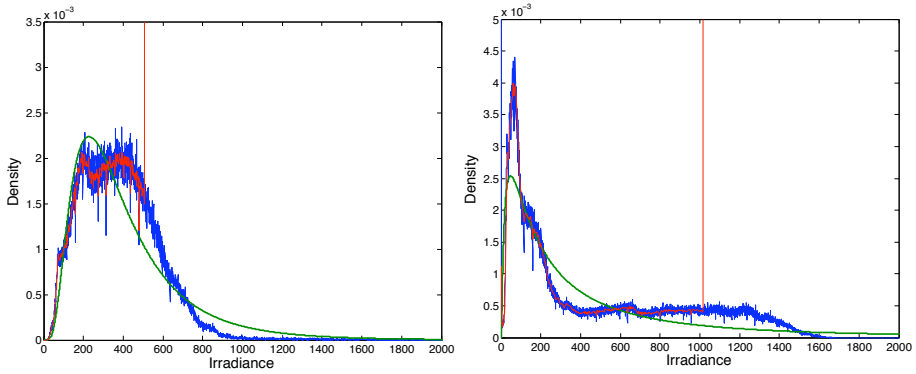
where  $\Phi_T$  is a quantity that decreases with  $T$  but *does not* depend on the density  $p(I)$ . For example, if  $L(e_T) = |e_T|^p$ , then  $\Phi_T = (\Delta/T)^2/12$  for  $p = 2$ ,  $\Phi_T = \Delta/4T$  for  $p = 1$ , and  $\Phi_T = \sqrt{2}\Delta/T/3$  for  $p = 0.5$ .

Eq. (8) formalizes a very intuitive concept, termed “Expose to the right” (ETTR) in the photography community [16]: increasing the exposure time improves the rendering quality for the non-saturated pixels. At the same time, increasing the exposure leads to more saturated pixels as well as to higher over-load error for the saturated pixels.

#### 4.1 Modeling the Irradiance Distribution

What is a good model for the density  $p(I)$ ? Suppose for a moment that all pixels in the image, taken at exposure  $T$ , are unsaturated. Let us define the “continuous domain” histogram as the piecewise constant function  $h_T(x)$  representing the proportion of pixels with  $B_T = \text{round}(x)$ . Note that  $h_T(2^N - 1)$  is the proportion of saturated pixels in the image. The continuous domain histogram  $h_T(x)$  can be used to model the density  $p(I)$  by means of the auxiliary function  $\bar{h}_T(I)$ , defined by (9) where  $f'$  is the derivative of  $f$ .

$$\bar{h}_T(I) = h_T(f(IT)/\Delta) \cdot f'(IT) \cdot T/\Delta \quad (9)$$



**Fig. 3.** The histogram function  $\bar{h}_T(I)$  for the “ground truth” 12-bit image (blue) and for a derived synthetic 8-bit image (red) are shown along with the lognormal density  $\bar{q}_T(I)$  fitted to the right-censored data from the 8-bit image for two different scenes. Note that the 8-bit images saturates for  $I = g((2^8 - 1)\Delta)/T$ .

But what if the image has saturated pixels? The brightness value of these pixels is not observed, and thus the histogram provides only partial information. For these case, we propose to model  $p(I)$  by means of a parametric function, with parameters learned from the unsaturated pixels. Parameter estimation from “right-censored” data is a well studied methodology, and standard methods exist [17,18]. In our experiments, we used the Matlab function `mle.m` which performs ML parameter estimation with right-censored data for a variety of parametric distributions.

We decided to use the lognormal parametric function for representing the marginal probability density function (pdf) of the irradiance data. This choice was suggested by the theoretical and experimental analysis of Richards [19] and Ruderman [20]. In particular, Richards [19] observed that random variables modeling distributions of important contributors to scene brightness, such as illumination sources and angles, surface reflectance, and the viewing angle for non-Lambertian surfaces, affect recorded brightness in a multiplicative fashion. Thus, the logarithm of brightness should be distributed as a sum of random variables, which the central limit theorem approximates as a normal distribution. It should be clear that any choice for a prior distribution of the brightness data is bound to fail in certain instances. For example, the presence of a strong illuminator, or even of the sky in an image, generates a peak in the brightness histogram that cannot be easily accounted for by a parametric distribution, especially if these peaks belong to the saturated region. Still, we believe that the chosen fit provides a simple and, in most cases, realistic estimation of the behavior of the irradiance even for the pixels that are saturated. An example of parametric fit is shown in Fig. 3 for two different scenes. Note that in both cases the 8-bit image saturates; the irradiance values for the saturated pixels are modeled by the lognormal fit.

Let  $q_T(B)$  be the parametric model learned from the right-censored brightness data taken with exposure  $T$ . Similarly to (9), a model  $\bar{q}_T(I)$  for  $p(I)$  based on  $q_T(B)$  can be defined as by:

$$\bar{q}_T(I) = q_T(f(IT)/\Delta) \cdot f'(IT)T/\Delta \quad (10)$$

At this point, we have two different representations for  $p(I)$ : the histogram-based function  $\bar{h}_T(I)$ , which is the best model for the unsaturated data; and the parametric density function  $\bar{q}_T(I)$ , which models the saturated and thus unobservable data. We propose a “composite” model  $\bar{p}_T(I)$  for  $p(I)$  that combines the two models above:

$$p(I) \approx \bar{p}_T(I) = \begin{cases} \bar{h}_T(I) & , I < g((2^N - 1)\Delta)/T \\ \bar{q}_T(I) / K_T & , I \geq g((2^N - 1)\Delta)/T \end{cases} \quad (11)$$

where  $K_T$  is a normalization constant:

$$K_T = h_T(2^N - 1) / \int_{g((2^N - 1)\Delta)/T}^{\infty} \bar{q}_T(I) dI \quad (12)$$

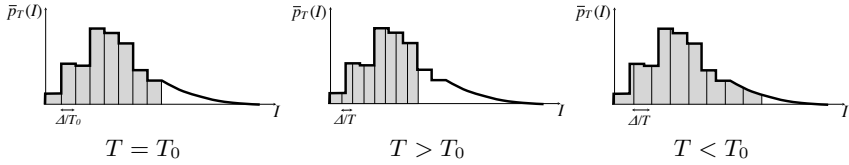
where we used the fact that  $h_T(2^N - 1)$  is the proportion of saturated pixels in the image. Basically, the image histogram is used to model  $p(I)$  for values of the radiant power  $I$  that do not generate saturation. For larger values (the “tail” part), the parametric model is used. Note that if all pixels are unsaturated, then the tail part of the density vanishes because  $K_T = 0$ . Note that, ideally,  $\bar{p}_T(I)$  should not change with  $T$ . The dependence of  $\bar{p}_T(I)$  on  $T$  is due to the fact that both histogram and fitting distribution are computed from a single image taken at exposure  $T$ .

Using the density  $\bar{p}_T(I)$  as an approximation to  $p(I)$ , one may compute the expected error  $E_T$  for a given image, taken at exposure  $T$ , as by (6). Note that, in the case of linear sensor characteristic, term  $\Phi_T$  in the expression (8) of the granular error component  $E_T^g$  can be pre-computed, as it does not depend on the data. The term  $\text{Prob}(I < 2^N \Delta/T)$  in (8) simply represents the portion of non-saturated pixels, and can be easily computed from the histogram. The overload error can be computed by integration of the parametric function  $q_T(I)$  via numerical or Monte Carlo methods.

## 5 Predicting the Optimal Exposure

In the previous section we showed how to estimate the expected rendering error for a given image. Now we extend our theory to the *prediction* of the expected error when  $T$  varies. Formally, we will try to predict the exposure error  $E_T$  at any value of  $T$  based on the observation of just one image taken with (known) exposure  $T_0$ . We will do so by modeling  $p(I)$  with our composite model  $\bar{p}_{T_0}(I)$  in (11). Then, the expected error at any value of exposure  $T$  can be estimated via (6).

Here are some details about our prediction algorithm (see also Fig. 4). We begin by considering values of  $T$  larger than  $T_0$ . The granular component  $E_T^g$  is easily computed from (7) or (8). The overload component  $E_T^o$  is equal to the sum of two terms. The first term represents the “projection” of the histogram  $\bar{h}_{T_0}$  into the overload area, that is, for  $I$  between  $g(2^N \Delta)/T$  and  $g(2^N \Delta)/T_0$ . Integration of  $L(e_T)\bar{p}_T(I)$  over this segment amounts to a sum using histogram values. The second term is obtained by integration of the error weighed by the parametric density  $\bar{q}_T(I)$  for values of  $I$  above  $g(2^N \Delta)/T_0$ . This term can be computed offline and stored in a look-up table for various parameters of the parametric function used.



**Fig. 4.** A representation of the composite density function  $\bar{p}_T(I)$ , under three different exposures. The shaded are represents the granular region. The area of the density within the shaded areas represents  $\text{Prob}(I < (2^N - 1)\Delta/T)$ .

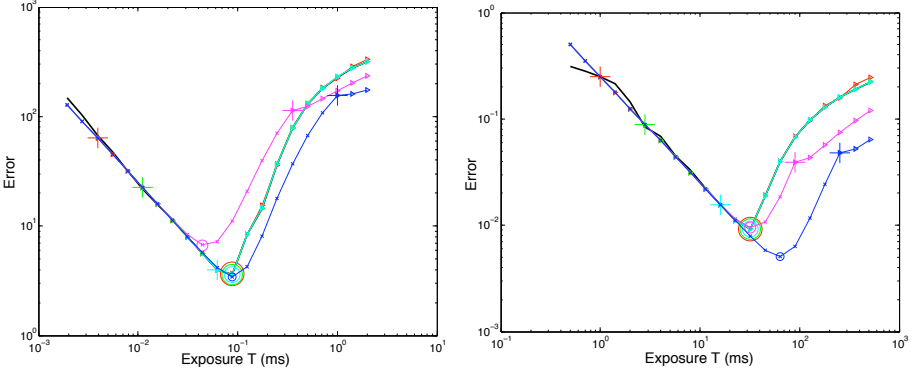
The predicted values for the granular and overload error components,  $\bar{E}_T^g$  and  $\bar{E}_T^o$ , can be expressed in a relatively simple form if the sensor’s characteristic  $f(I_T)$  is linear. In this case, the following identities hold:

$$\begin{aligned}
 T < T_0 : \quad & \bar{E}_T^g = \left[ (1 - h_{T_0}(2^N - 1)) + K_T \int_{(2^N - 1)/T_0}^{(2^N - 1)/T} \bar{q}_{T_0}(I) dI \right] \Phi_T \\
 & \bar{E}_T^o = K_T \int_{(2^N - 1)/T}^{\infty} L(I - (2^N - 1)/T) \bar{q}_{T_0}(I) dI \\
 T > T_0 : \quad & \bar{E}_T^g = \left[ \sum_{m=0}^{\text{floor}((2^N - 1)T_0/T)} h_{T_0}(m) \right] \Phi_T \\
 & \bar{E}_T^o = \sum_{m=\text{ceil}((2^N - 1)T_0/T)}^{2^N - 2} L(m/T_0 - (2^N - 1)/T) h_{T_0}(m) \\
 & \quad + K_T \int_{(2^N - 1)/T_0}^{\infty} L(I - (2^N - 1)/T) \bar{q}_{T_0}(I) dI
 \end{aligned} \tag{13}$$

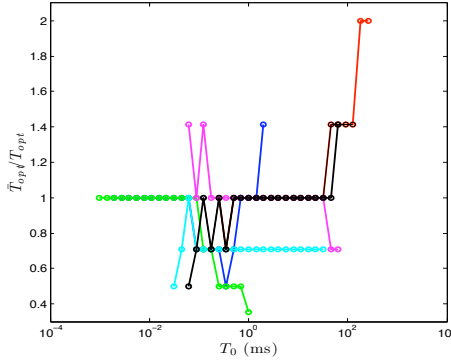
At this point, one may sample the estimated error  $\bar{E}_T = \bar{E}_T^g + \bar{E}_T^o$  for various values of  $T$  in order to find the estimated optimal exposure  $T_{\text{opt}}$ .

## 6 Experiments

We have used synthetically generated 8-bit images from a “ground truth” 12-bit image as discussed in the Appendix. The 12-bit images were taken with a Dragonfly 2 camera from Point Grey that has a very linear sensor characteristic  $f(I_T)$  [13][14][21]. The ground-truth 12-bit image is used for the computation of the ground-truth error  $E_T$  and of the optimal exposure  $T_{\text{opt}}$  that minimizes  $E_T$ .



**Fig. 5.** The ground-truth error  $E_T$  (black thick line) and the estimated errors  $\bar{E}_T$  starting from different values of  $T_0$  (thin colored lines, one line per choice of  $T_0$ ) for two different scenes. For each  $\bar{E}_T$  plot, the large '+' signs is placed at  $T_0$ : the whole plot is built from the analysis of the image at  $T_0$ . The large circles within each line represent the minimum of the plot, corresponding to the optimal exposure.



**Fig. 6.** Experiments with the proposed algorithm for estimating the optimal exposure  $T_{\text{opt}}$  from a single image. Each color represents a different scene. For each scene, the image exposed at  $T_0$  was used to find the estimate  $\bar{T}_{\text{opt}}$  using the algorithm in (13). The ratio  $\bar{T}_{\text{opt}}/T_{\text{opt}}$  is shown for each image with varying  $T_0$ . A value of  $\bar{T}_{\text{opt}}/T_{\text{opt}}$  equal to 1 means that the algorithm found the optimal exposure correctly.

Fig. 5 shows a number of estimated error plots  $\bar{E}_T$  as a function of exposure  $T$ . Each plot corresponds to a different starting point  $T_0$ . The thick black line is the “ground-truth” error  $E_T$ . Note that the left part of  $E_T$  has linear 45° slope in log-log space. This is because, for our choice of  $L(e_T) = |e_T|$ , the expected granular error is equal to  $\Delta/4T$  as mentioned in Sec. 4. However, for very small values of  $T$ , the granular error characteristic is not linear anymore, due to the fact that the “high rate” assumption does not hold true in these cases. The estimated error curves  $\bar{E}_T$  are generally good when the starting point  $T_0$  is in a location with few saturated pixels. The more challenging (and interesting)

situations are for larger  $T_0$ , chosen when a considerable portion of the image is saturated. In these cases, the estimated  $\bar{E}_T$  may fail to represent  $E_T$  in some areas, possibly leading to errors in the estimation of  $T_{\text{opt}}$ .

Results showing the quality of estimation of the optimal exposure from an image taken at exposure  $T_0$  for different values of the “start” exposure  $T_0$  are shown in Fig. 6 for various scenes. The optimal exposure  $T_{\text{opt}}$  for each scene was computed as discussed in Sec. 3. The plots in Fig. 6 show the ratio  $\bar{T}_{\text{opt}}/T_{\text{opt}}$ , which is indicative of the quality of the algorithm (values equal to 1 indicate correct estimation). Note that the different scenes had different optimal exposures  $T_{\text{opt}}$ . In most situations, our algorithm predicts the optimal exposure with good accuracy. However, when  $T_0$  is much smaller or higher than  $T_{\text{opt}}$ , the estimate may become incorrect. Small values of  $T_0$  mean that the histogram has little information due to high quantization step. Large values of  $T_0$  mean that the “start” image had a considerable number of saturated pixels.

## 7 Conclusion

We have presented a technique to estimate the optimal exposure from analysis of a single image. This approach relies on a definition of exposure quality based on the expected rendering error. Predicting the exposure quality for varying exposure times requires accessing the saturated (and thus unobservable) pixels. We proposed the use of a parametric distribution that fits the observable data, and allows reasoning about the saturated data. Our experiments show that this model enables accurate one-shot estimation of the correct exposure as long as the image being analyzed does not contain too many saturated pixels, or is not too under-exposed.

One main limitation of our approach is that we do not consider sensor noise and the use of gain as an exposure parameter. Future work will address both these issues, along with the possibility of using more accurate models for the distribution of irradiance in the image. Eventually, our algorithm will be integrated in a dynamic loop for real-time exposure control in video applications.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. BES-0529435.

## References

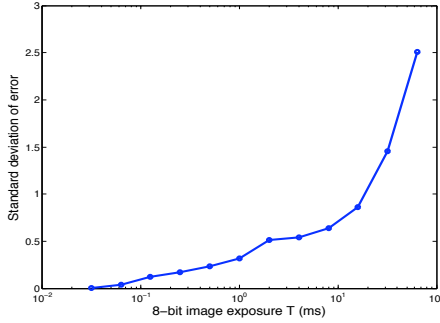
1. Muramatsu, M.: Photometry device for a camera (1997)
2. Johnson, B.K.: Photographic exposure control system and method (1997)
3. Kremens, R., Sampat, N., Venkataraman, S., Yeh, T.: System implications of implementing auto-exposure on consumer digital cameras. In: Proceedings of the SPIE Electronic Imaging Conference, vol. 3650 (1999)

4. Shimizu, S., Kondo, T., Kohashi, T., Tsuruta, M., Komuro, T.: A new algorithm for exposure control based on fuzzy logic for video cameras. *IEEE Transactions on Consumer Electronics* 38, 617–623 (1992)
5. Yang, M., Crenshaw, J., Augustine, B., Mareachen, R., Wu, Y.: Face detection for automatic exposure control in handheld camera. In: *IEEE International Conference on Computer Vision Systems* (2006)
6. Nuske, S., Roberts, J., Wyeth, G.: Extending the range of robotic vision. In: *IEEE International Conference on Robotics and Automation* (2006)
7. Nourani-Vatani, N., Roberts, J.: Automatic exposure control. In: *Australasian Conference on Robotics and Automation* (2007)
8. Kuno, T., Matoba, N.: A new automatic exposure system for digital still cameras. *IEEE Transactions on Consumer Electronics* 44, 192–199 (1998)
9. Nayar, S., Branzoi, V.: Adaptive dynamic range imaging: optical control of pixel exposures over space and time. In: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1168–1175 (2003)
10. Schulz, S., Grimm, M., Grigat, R.R.: Optimum auto exposure based on high-dynamic-range histogram. In: *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA'07)*, Stevens Point, Wisconsin, USA, pp. 269–274. World Scientific and Engineering Academy and Society, WSEAS (2007)
11. Grossberg, M., Nayar, S.: High dynamic range from multiple images: Which exposures to combine? In: *Proceedings of the ICCV Workshop on Color and Photometric Methods in Computer Vision, CPMCV* (2003)
12. Barakat, N., Hone, A., Darcie, T.: Minimal-bracketing sets for high-dynamic-range image capture. *IEEE Transactions on Image Processing* 17, 1864–1875 (2008)
13. Mitsunaga, T., Nayar, S.K.: Radiometric self calibration. *Proceedings of the IEEE Computer Vision and Pattern Recognition* 1, 1374 (1999)
14. Matsushita, Y., Lin, S.: Radiometric calibration from noise distributions. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07* (2007)
15. Gersho, A., Gray, R.: *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell (1991)
16. Anon, E., Grey, T.: *Photoshop for Nature Photographers: A Workshop in a Book*. Wiley, Chichester (2005)
17. Miller, R., Gong, G., Muñoz, A.: *Survival analysis*. Wiley, New York (1981)
18. Gross, Shulamith, T., LaiTze, L.: Nonparametric estimation and regression analysis with left-truncated and right-censored data. *Journal of the American Statistical Association* 91, 1166–1180 (1996)
19. Richards, W.: Lightness scale from image intensity distributions. *Applied Optics* 21, 2569–2582 (1982)
20. Ruderman, D.: The statistics of natural images. *Network: computation in neural systems* 5, 517–548 (1994)
21. Point Grey Research, Inc. *Point Grey Dragonfly2 Technical Specification* (2007)
22. Chen, T., El Gamal, A.: Optimal scheduling of capture times in a multiple capture imaging system. In: *Proc. SPIE*, vol. 4669, pp. 288–296 (2002)

## Appendix A

This Appendix describes the process used to generate synthetic 8-bit images at different exposure  $T$  starting from a 12-bit image. We used a Dragonfly 2 camera





**Fig. 7.** The standard deviation of the error  $B_{T,8} - B_{T,12}$  between the synthetic and the real 8-bit images as a function of the exposure  $T$

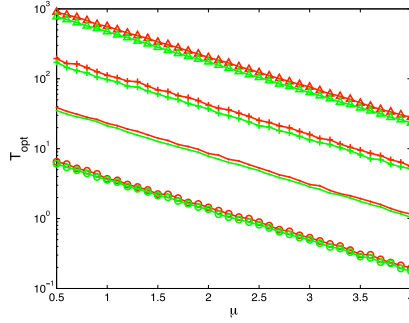
from Point Grey that has a very linear sensor characteristic  $f(I_T)$  [13,14,21] and provides images both at 12-bit and 8-bit pixel depth. Images were taken at 12 bits, carefully choosing the exposure  $T_0$  so as to best exploit the camera’s dynamic range while avoiding saturation. Images with more than 0.1% pixels saturated were discarded. The brightness data  $B_{T_0,12}$  was dithered by adding white noise with uniform distribution between 0.5 and 0.5, then divided by  $2^{12-8} = 16$ . This quantity is multiplied by  $T_0/T$  and then quantized with  $\Delta = 1$  in order to obtain the equivalent 8-bit image for exposure  $T$ , named  $B_{T,12}$ . In this way, multiple 8-bit synthetic images can be obtained for different exposure value  $T$ .

In order to assess the error that should be expected with this processing, we took a number of real 8-bit images ( $B_{T,8}$ ) of a static scene with various exposures  $T$ , and then compared them with their synthetic counterparts obtained by synthesis from a 12-bit image of the same scene. The results, in terms of standard deviation of the error  $B_{T,8} - B_{T,12}$ , are plotted in Fig. 7. As expected, the error increases with increasing exposure  $T$  (remember that the dithered 12-bit image is multiplied by  $T/T_0$ ). Note that for most of the exposure, the standard deviation stays below 1 (PSNR = 48 dB), and it reaches a maximum of about 2.5 (PSNR = 40 dB).

## Appendix B

In this Appendix we consider the effect of photon noise in the determination of the optimal exposure. For a given value of irradiant power  $I$  and of exposure  $T$ , the variance of the rendering error due to photon noise is equal to  $\sigma_{\text{pht}}^2 = qI/T$ , where  $q$  is the electrical charge of an electron, and  $I$  is measured in terms of photoelectronic current [22]. Let  $N_{\text{sat}}$  be the full well capacity of the sensor. It is reasonable to assume that (in the absence of amplification gain), the quantizer saturates when the sensor saturates, that is,  $\Delta(2^N - 1) = qN_{\text{sat}}$ .

When computing the optimal exposure, both quantization and photon noise should be considered. Unfortunately, the resulting rendering error depends on



**Fig. 8.** Monte Carlo simulation of  $T_{\text{opt}}$  for  $L(e_T) = |e_T|$ , red: photon noise and quantization noise considered, green: quantization noise only, triangles:  $\sigma = 0.5$ , plus marks:  $\sigma = 1$ , dots:  $\sigma = 1.5$ , circles:  $\sigma = 2.0$

the characteristics of the irradiance distribution. For example, one can easily derive the expression of the quadratic norm of the granular error under the assumption of linear sensor characteristic:

$$E_T^g = \frac{q^2 N_{\text{sat}}}{T^2} \left( \frac{N_{\text{sat}}}{12(2^N - 1)^2} + \frac{TE[I]}{qN_{\text{sat}}} \right) \quad (14)$$

where  $E[I]$  is the average value of the irradiant power. The second term within the parenthesis is a number that represents the “average degree of saturation”. In particular, when no pixel is saturated, then  $TE[I]/qN_{\text{sat}} < 1$ . Note that for (14), the *relative* effect of the term due to photon noise is increased as  $N_{\text{sat}}$  decreases. Unfortunately, computation of the average error under different metrics (in particular,  $L(e_T) = |e_T|$ , which is the metric considered in this paper) requires knowledge of the probability density function of the irradiance  $I$ .

Fig. 8 shows results of a Monte Carlo simulation to find  $T_{\text{opt}}$  for  $L(e_T) = |e_T|$ , assuming a log-normal distribution with parameters  $\mu$  and  $\sigma$ . Fixed parameters in the simulation are  $N_{\text{sat}} = 6000$  (representing a sensor with a relatively small well capacity) and bit depth  $N = 8$ . Two million points are sampled to generate error values for each  $T$  used in the search for  $T_{\text{opt}}$ . Results shown in Fig. 8 suggest that the ratio of  $T_{\text{opt}}$  with photon noise considered, relative to  $T_{\text{opt}}$  where it is not, is a small positive value.  $T_{\text{opt}}$  appears more sensitive to a choice of  $L$  or change in the irradiance distribution at the sensor than to the consideration of photon noise.

# Analyzing Depth from Coded Aperture Sets

Anat Levin

Dep. of Computer Science and Applied Math, The Weizmann Institute of Science

**Abstract.** Computational depth estimation is a central task in computer vision and graphics. A large variety of strategies have been introduced in the past relying on viewpoint variations, defocus changes and general aperture codes. However, the tradeoffs between such designs are not well understood. Depth estimation from computational camera measurements is a highly non-linear process and therefore most research attempts to evaluate depth estimation strategies rely on numerical simulations. Previous attempts to design computational cameras with good depth discrimination optimized highly non-linear and non-convex scores, and hence it is not clear if the constructed designs are optimal. In this paper we address the problem of depth discrimination from  $J$  images captured using  $J$  arbitrary codes placed within one fixed lens aperture. We analyze the desired properties of discriminative codes under a geometric optics model and propose an *upper bound* on the best possible discrimination. We show that under a multiplicative noise model, the half ring codes discovered by Zhou et al. [1] are near-optimal. When a large number of images are allowed, a multi-aperture camera [2] dividing the aperture into multiple annular rings provides near-optimal discrimination. In contrast, the plenoptic camera of [5] which divides the aperture into compact support circles can achieve at most 50% of the optimal discrimination bound.

## 1 Introduction

Estimating scene depth from image measurements is a central goal of computer vision research. Historical depth estimation strategies utilize viewpoint or defocus cues. This includes stereo [3] and plenoptic cameras [4, 5], depth from focus and depth from defocus techniques [2, 6]. Recent computational cameras combine and extend these strategies using coded apertures [1, 7, 8] and phase masks [9, 10].

The large variety of computational cameras calls for a systematic way to compare their performance and understand limitations. However, despite the large amount of research on the subject, the problem is far from being understood. Historical comparisons between stereo and DFD approaches have attracted a lot of research [11–13], leading to different conclusions based on different experimental setups. An important analytic analysis is proposed by Schechner and Kiryati [11], who point that to compare the two approaches the same physical dimensions should be used, and the stereo baseline should be equal to the lens aperture. Despite the important contribution of their analysis, many open questions remain. In particular, they model noise sensitivity on a per-frequency basis and do not analyze how the discrimination information is combined over multiple frequencies.

The problem of computational camera analysis is gaining increased attention [1, 10, 11, 14–17]. Recently, success has been achieved in understanding the related problem

of removing defocus blur given depth [10, 16, 17]. However, while the accuracy of depth estimation has an important effect on the quality of defocus deblurring, depth discrimination accuracy is often omitted from the analysis [10, 17], or evaluated numerically only [14, 16]. Some authors [1, 8] address the depth discrimination problem directly and propose explicit scores for discrimination accuracy. However, depth discrimination is a non linear process and the proposed discrimination scores are highly non linear as well. Specifically, Zhou et al. [1] searched for discriminative codes using a genetic algorithm. While the discovered codes are interesting, it is not clear if they are optimal since there is no way to test whether the global optimum of the discrimination score was reached by the optimization algorithm. This optimization also did not offer concrete understanding of the characteristics of good codes.

In this manuscript we address depth discrimination in a setting similar to [1]. One is allowed to capture  $J \geq 2$  images of a scene using  $J$  different aperture code masks. All images, however, are taken by a fixed static camera, using a standard lens at a fixed focus setting. The aperture codes are allowed to have any general shape within the maximal lens aperture width. This problem formulation is general enough to cover most existing depth estimation strategies. For example, depth from defocus can be expressed using disc aperture masks with different widths, and stereo can be expressed using code masks with holes allowing light at opposite ends of the aperture. We note that in this setting all designs have the same physical dimensions since they are bounded within the same maximal aperture which provides an upper bound on their discrimination performance. We restrict the discussion to the geometric optics model and our results are valid only up to the extent at which this model is valid.

We ask what is the quality of depth discrimination that a given codes set can provide, and what are the best results one can hope to achieve. We build on the discrimination score derived by [1] but notice that it can be analyzed analytically using the derivatives of the code spectra, and larger variations improve discrimination. We use Parseval's theorem and the fact that the primal support of the codes is bounded to show that the maximal derivative power is bounded. This analysis allows us to derive an analytic *upper bound* on the maximal possible depth discrimination. It also provides an understanding of the desired properties of good aperture codes. We use this to analyze existing depth discrimination strategies. For the case of  $J = 2$  images and multiplicative noise, we show that the half ring codes discovered by [1] are indeed near optimal. When a large number  $J$  of images is allowed, near optimal discrimination can be obtained when the aperture is divided to multiple annular rings, along the lines of the multi-aperture camera of [2]. In contrast, dividing the aperture into small compact squares or circles as done by the plenoptic camera of [5] can achieve no more than 50% of the discrimination bound.

## 2 Depth Discrimination from Coded Apertures

**Problem formulation:** We consider a camera with a standard lens focused at depth  $d_0$ .  $J$  images are captured via  $J$  codes which block light at different regions of the aperture. The codes can be expressed as  $J$  functions  $a^1, \dots, a^J$  bounded within a maximal aperture radius  $R$ . That is:  $\forall x, y \ 0 \leq a_{(x,y)}^j \leq 1$ ,  $a_{(x,y)}^j = 0$  if  $|x|^2 + |y|^2 > R^2$ . We

also restrict the  $J$  codes to disjoint parts of the aperture so that the  $J$  images can be captured simultaneously (e.g. [2, 5]). That is, for each  $x, y$  there is a single  $j$  for which  $a_{(x,y)}^j > 0$ . However, many of our results hold without the disjointness requirement.

Let  $I$  denote an ideal sharp version of a scene, and assume we observe  $J$  images  $B^1, \dots, B^J$ . If the depth is locally constant the observed images of an object at depth  $d$  can be described as a local convolution of the sharp version  $I$  with Point Spread Functions (PSFs)  $\phi^d$ . In the frequency domain the imaging is expressed as multiplication with an Optical Transfer Function (OTF)  $\hat{\phi}^d$ :

$$\hat{B}_{\omega_{x,y}}^j = \hat{\phi}_{\omega_{x,y}}^{d,j} \hat{I}_{\omega_{x,y}} + n_{\omega_{x,y}}^j, \quad (1)$$

where  $\omega_{x,y} = (\omega_x, \omega_y)$  denote spatial frequency coordinates and  $n^j$  is an imaging noise. Through this paper we index depth using the corresponding light field slope  $s = (d - d_0)/d$ , since the PSF and OTF vary as a linear function of  $s$  and not of  $d$ . Using a geometric optics model, it was shown [7, 8] that the PSFs and OTFs are scaled versions of the aperture codes:

$$\hat{\phi}_{\omega_{x,y}}^{s,j} = \hat{a}_{(s,\omega_{x,y})}^j \quad (2)$$

The scaled PSF model, however, does not take into account wave optics effects. The geometric optics model is a reasonable approximation to the true optics when the holes in the code are not too small. The optimality arguments in this paper are only valid to the extent at which the geometric model is valid.

**Noise model:** We follow the affine noise model (e.g. [16]). For simplicity this model assumes the noise is a zero mean Gaussian whose variance  $\eta^2$  is constant over the image, and it is a combination of a constant additive term, the read noise, and a multiplicative term, the photon noise:

$$(\eta^j)^2 = \alpha^j \eta_{mult}^2 + \eta_{add}^2, \quad (3)$$

where  $\alpha^j$  is the amount of light gathered by the  $j$ 'th aperture  $\alpha^j = \iint a^j d\omega_x d\omega_y$ . For modern sensors under good illumination conditions the noise is dominated mostly by the multiplicative term and  $(\eta^j)^2 \approx \alpha^j \eta_{mult}^2$ . We often assume that all  $J$  apertures have equal area and omit the  $j$  index from  $\alpha, \eta$ .

Given a set of observed images  $B^1, \dots, B^J$ , estimating the scene depth is equivalent to estimating at every local image window a slope index  $s$  such that locally  $B^j$  was blurred with  $\phi^{s,j}$ . Our goal in this paper is to analyze the quality of depth discrimination that a set of aperture codes can provide. We note that all codes are bounded within the same maximal aperture radius  $R$  and the maximal radius is the parameter upper-bounding the performance of all codes. We start with a brief review of the depth discrimination score proposed in [1]. In Section 3, we derive bounds on this discrimination score and study the desired properties of optimal discrimination codes.

## 2.1 Desircrimination Score

One often assumes a zero mean Gaussian prior on the sharp signal  $\hat{I}$ . For simplicity the classical  $1/f^2$  law is used and the variance in frequency  $\omega_{x,y}$  is set to  $\sigma_{\omega_{x,y}}^2 = 1/|\omega_{x,y}|^2$ . We denote with bold fonts the  $J$  dimensional vectors  $\hat{\mathbf{B}}_{\omega_{x,y}} = [\hat{B}_{\omega_{x,y}}^1, \dots, \hat{B}_{\omega_{x,y}}^J]$ ,  $\hat{\boldsymbol{\phi}}_{\omega_{x,y}}^s = [\hat{\phi}_{\omega_{x,y}}^{s,1}, \dots, \hat{\phi}_{\omega_{x,y}}^{s,J}]$ ,  $\hat{\mathbf{a}}_{\omega_{x,y}} = [\hat{a}_{\omega_{x,y}}^1, \dots, \hat{a}_{\omega_{x,y}}^J]$ . The

<sup>1</sup> Through this manuscript we use  $\hat{\cdot}$  to denote the Fourier transform of the corresponding signal.

probability of the observed images factorizes as an independent product over individual frequencies  $P(\hat{B}^1, \dots, \hat{B}^J) = \prod_{\omega_{x,y}} P(\hat{\mathbf{B}}_{\omega_{x,y}})$ . Since  $\hat{\mathbf{B}}_{\omega_{x,y}}$  is obtained from  $\hat{I}$  as a linear transformation plus Gaussian noise,  $P(\hat{\mathbf{B}}_{\omega_{x,y}})$  follows a Gaussian distribution with covariance  $\Psi_{\omega_{x,y}}^s = \hat{\phi}_{\omega_{x,y}}^s \sigma_{\omega_{x,y}}^2 \hat{\phi}_{\omega_{x,y}}^{s*} + \eta^2 \mathbb{I}$ , where  $\mathbb{I}$  denotes a  $J \times J$  identity matrix and  $*$  denotes the conjugate transpose. Let  $U_{\omega_{x,y}}^s$  be a  $J \times (J-1)$  matrix completing  $\hat{\phi}_{\omega_{x,y}}^s$  to an orthogonal basis, that is, the  $J \times J$  matrix  $\tilde{U}_{\omega_{x,y}}^s = \begin{bmatrix} \hat{\phi}_{\omega_{x,y}}^s \\ |\hat{\phi}_{\omega_{x,y}}^s| \end{bmatrix}, U_{\omega_{x,y}}^s$  is orthogonal. We can then express:

$$\Psi_{\omega_{x,y}}^s = \tilde{U}_{\omega_{x,y}}^s D \left( |\hat{\phi}_{\omega_{x,y}}^s|^2 \sigma_{\omega_{x,y}}^2 + \eta^2, \eta^2, \dots, \eta^2 \right) \tilde{U}_{\omega_{x,y}}^{s*}, \quad (4)$$

$$\Psi_{\omega_{x,y}}^{s-1} = \tilde{U}_{\omega_{x,y}}^s D \left( \frac{1}{|\hat{\phi}_{\omega_{x,y}}^s|^2 \sigma_{\omega_{x,y}}^2 + \eta^2}, \frac{1}{\eta^2}, \dots, \frac{1}{\eta^2} \right) \tilde{U}_{\omega_{x,y}}^{s*}, \quad (5)$$

where  $D(\dots)$  denotes a diagonal matrix. We assume the signal variance is sufficiently above the noise level:  $|\hat{\phi}_{\omega_{x,y}}^s|^2 \sigma_{\omega_{x,y}}^2 > \eta^2$  (other frequencies provide little discrimination and their contribution can be ignored).  $\Psi$  allows high variance along the OTFs direction  $\hat{\phi}_{\omega_{x,y}}^s$  and low variance at all orthogonal directions. The expected negative log likelihood of an observation whose correct depth is  $s_0$  under possible explanation  $s$  is:

$$E_{P(B|s_0)}[-2 \log P(B|s)] = E_{P(B|s_0)}[B^* \Psi^s B] + \log |\Psi^s| \quad (6)$$

$$\begin{aligned} &= \sum_{\omega_{x,y}} \left[ \frac{|\hat{\phi}_{\omega_{x,y}}^{s_0}|^2 \sigma_{\omega_{x,y}}^2 + \eta^2}{|\hat{\phi}_{\omega_{x,y}}^{s_0}|^2 \eta^2} |U_{\omega_{x,y}}^{s*} \hat{\phi}_{\omega_{x,y}}^{s_0}|^2 + \frac{\eta^2}{|\hat{\phi}_{\omega_{x,y}}^s|^2 (|\hat{\phi}_{\omega_{x,y}}^s|^2 \sigma^2 + \eta^2)} |\hat{\phi}_{\omega_{x,y}}^{s*} U_{\omega_{x,y}}^{s_0}|^2 \right. \\ &\quad \left. + \frac{|\hat{\phi}_{\omega_{x,y}}^{s_0}|^2 \sigma_{\omega_{x,y}}^2 + \eta^2}{|\hat{\phi}_{\omega_{x,y}}^s|^2 |\hat{\phi}_{\omega_{x,y}}^{s_0}|^2 (|\hat{\phi}_{\omega_{x,y}}^s|^2 \sigma_{\omega_{x,y}}^2 + \eta^2)} |\hat{\phi}_{\omega_{x,y}}^{s*} \hat{\phi}_{\omega_{x,y}}^{s_0}|^2 + |U_{\omega_{x,y}}^{s*} U_{\omega_{x,y}}^{s_0}|^2 + \log |\Psi_{\omega_{x,y}}^s| \right] \quad (7) \\ &\approx \sum_{\omega_{x,y}} \frac{\sigma_{\omega_{x,y}}^2}{\eta^2} |U_{\omega_{x,y}}^{s*} \hat{\phi}_{\omega_{x,y}}^{s_0}|^2 \quad (8) \end{aligned}$$

where the approximation of Eq. (8) follows from the fact that Eq. (7) is dominated by the first term when the noise is small relative to the signal, and the  $\log |\Psi_{\omega_{x,y}}^s|$  term is relatively constant. That means that a good discrimination is obtained when  $\hat{\phi}^s$  and  $\hat{\phi}^{s_0}$  are as orthogonal as possible.

The discrimination score in Eq. (8) is a simple extension of the one derived by [1] from the two image case to the  $J$  image case. In [1] the discrimination score was evaluated discretely over a sample of  $s$  values. Since discrimination is usually most challenging at the neighborhood of the true solution<sup>2</sup> we propose to replace the discrete sample with an analytic derivative of the OTF as a function of depth:  $\nabla_s \hat{\phi}_{\omega_{x,y}}^s = \frac{\partial \hat{\phi}_{\omega_{x,y}}^s}{\partial s}$  ( $\nabla_s \hat{\phi}_{\omega_{x,y}}^s$  is a  $J$ -dimensional vector). We note that since  $U_{\omega_{x,y}}^{s*} \hat{\phi}_{\omega_{x,y}}^s = 0$ ,  $U_{\omega_{x,y}}^{s*} \hat{\phi}_{\omega_{x,y}}^{s_0} = U_{\omega_{x,y}}^{s*} (\hat{\phi}_{\omega_{x,y}}^{s_0} - \hat{\phi}_{\omega_{x,y}}^s) \approx U_{\omega_{x,y}}^{s*} \nabla_s \hat{\phi}_{\omega_{x,y}}^s$ . We denote by  $\mathcal{D}_{\omega_{x,y}}^s(\hat{\phi})$  the local discrimination score at frequency  $\omega_{x,y}$ :

$$\mathcal{D}_{\omega_{x,y}}^s(\hat{\phi}) = \frac{\sigma_{\omega_{x,y}}^2}{\eta^2} |U_{\omega_{x,y}}^{s*} \nabla_s \hat{\phi}_{\omega_{x,y}}^s|^2 = \frac{\sigma_{\omega_{x,y}}^2}{\eta^2} \left( |\nabla_s \hat{\phi}_{\omega_{x,y}}^s|^2 - \frac{1}{|\hat{\phi}_{\omega_{x,y}}^s|^2} |\hat{\phi}_{\omega_{x,y}}^{s*} \nabla_s \hat{\phi}_{\omega_{x,y}}^s|^2 \right), \quad (9)$$

<sup>2</sup> This model does not penalize the symmetry of the PSF in front and behind the focus depth.

which implies that discrimination is maximized when there is a large variation of the OTFs as a function of  $s$ , in the direction orthogonal to  $\hat{\phi}^s$ . We wish to find OTFs maximizing discrimination integrated over all frequencies (up to spatial resolution  $\Omega$ ):

$$\mathfrak{D}^s(\hat{\phi}) = \int_{-\Omega}^{\Omega} \int_{-\Omega}^{\Omega} \mathfrak{D}_{\omega_x, y}^s(\hat{\phi}) d\omega_x d\omega_y. \quad (10)$$

### 3 Discrimination Budget

To understand how to maximize the discrimination score we study some of the physical constraints on the OTF. Let  $a^j$ ,  $\hat{a}^j$  denote the aperture code of the  $j$ 'th view and its Fourier transform. The PSF and OTF are known to be scaled versions of the aperture code [7, 8]:  $\hat{\phi}_{\omega_x, y}^{s, j} = \hat{a}_{(s \cdot \omega_x, y)}^j$ . This implies that

$$\nabla_s \hat{\phi}_{\omega_x, y}^{s, j} = |\omega_{x, y}| \nabla_{\omega_{x, y}} \hat{a}_{(s \omega_{x, y})}^j = |\omega_{x, y}| \left( \frac{\omega_x}{|\omega_{x, y}|} \frac{\partial \hat{a}^j}{\partial \omega_x} + \frac{\omega_y}{|\omega_{x, y}|} \frac{\partial \hat{a}^j}{\partial \omega_y} \right). \quad (11)$$

The  $|\omega_{x, y}|$  factor multiplying the derivative in Eq. (11) is canceled by  $\sigma_{\omega_{x, y}}^2 = 1/|\omega_{x, y}|^2$  in Eq. (9), and the discrimination score of Eq. (9) can be expressed as a function of the derivatives of the aperture spectra  $\hat{a}^j$  weighting all entries equally:

**Definition 1.** Consider a set of aperture codes  $\hat{\mathbf{a}}$ . The *depth discrimination score at spatial frequency*  $\omega_{x, y}$  is defined as

$$\mathfrak{D}_{\omega_{x, y}}(\hat{\mathbf{a}}) = \frac{1}{\eta^2} \left( |\nabla_{\omega_{x, y}} \hat{\mathbf{a}}_{\omega_{x, y}}|^2 - \frac{|\hat{\mathbf{a}}_{\omega_{x, y}}^* \nabla_{\omega_{x, y}} \hat{\mathbf{a}}_{\omega_{x, y}}|^2}{|\hat{\mathbf{a}}_{\omega_{x, y}}|^2} \right) \quad (12)$$

and the *total discrimination score* at depth  $s$  as

$$\mathfrak{D}^s(\hat{\mathbf{a}}) = \frac{1}{s^2} \int_{-s\Omega}^{s\Omega} \int_{-s\Omega}^{s\Omega} \mathfrak{D}_{\omega_{x, y}}(\hat{\mathbf{a}}) d\omega_x d\omega_y. \quad (13)$$

We note that the discrimination around depth  $s$  is integrated up to a cut-off frequency  $s\Omega$  but multiplied by the density  $1/s^2$ . We often omit the  $s$  index from Eq. (13) and consider the case  $s = 1$  which is usually the more challenging case. (the  $1/s^2$  factor improves the score of low  $s$  values and in addition, there is usually more energy at the low frequencies).

The discrimination score cannot be made arbitrarily large. Our goal is to show that the best possible discrimination score is bounded and then understand the desired properties of aperture codes with optimal discrimination. We define two useful quantities:

**Definition 2.** The *center-oriented derivative power at frequency*  $\omega_{x, y}$  is

$$\mathfrak{C}_{\omega_{x, y}}(\hat{\mathbf{a}}) = \frac{1}{\eta^2} \sum_j \left| \nabla_{\omega_{x, y}} \hat{a}_{\omega_{x, y}}^j \right|^2 = \frac{1}{\eta^2} \sum_j \left| \frac{\omega_x}{|\omega_{x, y}|} \frac{\partial \hat{a}^j}{\partial \omega_x} + \frac{\omega_y}{|\omega_{x, y}|} \frac{\partial \hat{a}^j}{\partial \omega_y} \right|^2, \quad (14)$$

the *total center-oriented derivative power* is  $\mathfrak{C}^s(\hat{\mathbf{a}}) = \frac{1}{s^2} \int_{-s\Omega}^{s\Omega} \int_{-s\Omega}^{s\Omega} \mathfrak{C}_{\omega_{x, y}}(\hat{\mathbf{a}}) d\omega_x d\omega_y$ .

**Definition 3.** The *gradient power at frequency*  $\omega_{x,y}$  is

$$\mathfrak{G}_{\omega_{x,y}}(\hat{\mathbf{a}}) = \frac{1}{\eta^2} \sum_j \left| \nabla \hat{a}_{\omega_{x,y}}^j \right|^2 = \frac{1}{\eta^2} \sum_j \left[ \left| \frac{\partial \hat{a}^j}{\partial \omega_x} \right|^2 + \left| \frac{\partial \hat{a}^j}{\partial \omega_y} \right|^2 \right] \quad (15)$$

and the *total gradient power* is  $\mathfrak{G}^s(\hat{\mathbf{a}}) = \frac{1}{s^2} \int_{-s\Omega}^{s\Omega} \int_{-s\Omega}^{s\Omega} \mathfrak{G}_{\omega_{x,y}}(\hat{\mathbf{a}}) d\omega_x d\omega_y$ .

Note that while we use the terms derivative and gradient, the quantities in Eqs. (14) and (15) are actually normalized by the noise variance  $\eta^2$ . The definition of the discrimination score in Eq. (12) implies that it is bounded by the center-oriented derivative power  $\mathfrak{D}_{\omega_{x,y}}(\hat{\mathbf{a}}) \leq \mathfrak{C}_{\omega_{x,y}}(\hat{\mathbf{a}})$ . Also the oriented derivative power is bounded by the gradient power  $\mathfrak{C}_{\omega_{x,y}}(\hat{\mathbf{a}}) \leq \mathfrak{G}_{\omega_{x,y}}(\hat{\mathbf{a}})$ . We observe that the total gradient power cannot be made arbitrarily high. We show that there is a fixed energy budget which is determined by the primal support of  $a^j$ , and this budget is preserved in the frequency domain as a consequence of Parseval's theorem. Bounding the gradient power provides a bound on the best possible discrimination score.

**Claim 1.** Let  $a^1, \dots, a^J$  be a set of disjoint codes inside an aperture of radius  $R$ . Their total gradient power is bounded and satisfies

$$\sum_{j=1}^J \frac{1}{\eta^2} \iint \left| \nabla \hat{a}_{\omega_{x,y}}^j \right|^2 d\omega_x d\omega_y = \sum_{j=1}^J \frac{1}{\eta^2} \iint \left| a_{(x,y)}^j \right|^2 (x^2 + y^2) dx dy \leq \mathfrak{B} \quad (16)$$

for

$$\mathfrak{B} = \max_{0 < \lambda < R} \frac{\frac{\pi}{2} (R^4 - (R - \lambda)^4)}{\frac{\pi}{2} (R^2 - (R - \lambda)^2) \eta_{mult}^2 + \eta_{add}^2} \quad (17)$$

*Proof.* Differentiating  $\hat{a}^j$  can be expressed as a convolution of  $\hat{a}^j$  with derivative filters  $f_{\omega_x}$ , such that  $\nabla_{\omega_x} \hat{a}^j = f_{\omega_x} \otimes \hat{a}^j$ . In the primal domain this convolution translates to multiplication  $f_{(x,y)} \cdot a_{(x,y)}^j$ . The Fourier transform of an ideal derivative filter is  $|\hat{f}_{(x,y)}| = |x|$ . Parseval's theorem implies that the frequency derivatives power is preserved in the primal domain, and thus:

$$\iint \left| f_{\omega_x} \otimes \hat{a}^j \right|^2 d\omega_x d\omega_y = \iint \left| a_{(x,y)}^j \right|^2 x^2 dx dy. \quad (18)$$

A similar property applies for  $\nabla_{\omega_y} \hat{a}^j$ . Therefore

$$\frac{1}{\eta^2} \iint \left| \nabla \hat{a}_{\omega_{x,y}}^j \right|^2 d\omega_x d\omega_y = \frac{1}{\eta^2} \iint \left| a_{(x,y)}^j \right|^2 (x^2 + y^2) dx dy. \quad (19)$$

We now ask what is the maximal value that the RHS of Eq. (19) can obtain. If  $a^j$  is open over a large area the integral value is increased, but if  $\eta_{mult}^2 > 0$ , a larger aperture area also increases the noise (see Eq. (3)). Thus, the optimal aperture area should trade discrimination v.s. noise, and depends on the ratio between  $\eta_{mult}^2$  to  $\eta_{add}^2$ . However, for every fix aperture area, we can ask what is the maximal value of the integral in the RHS of Eq. (19), among all codes with the same fixed area. The highest value is obtained by a ring attached to the aperture boundaries, since the  $x^2 + y^2$  values averaged in Eq. (19) are large when they are adjacent to the aperture boundary. Let  $r^\lambda$  denote a code open at



an outer ring of width  $\lambda$ ,  $r_{(x,y)}^\lambda = 1$  iff  $(R - \lambda)^2 \leq x^2 + y^2 \leq R^2$ . The gradient power of the code  $r^\lambda$  is the highest among all codes with the same area. Standard calculus implies that the area and gradient power of a ring  $r^\lambda$  are

$$\iint r_{(x,y)}^\lambda dx dy = \pi (R^2 - (R - \lambda)^2), \quad \iint r_{(x,y)}^\lambda (x^2 + y^2) dx dy = \frac{\pi}{2} (R^4 - (R - \lambda)^4). \quad (20)$$

Therefore the gradient power of a code is bounded by:

$$\frac{1}{\eta^2} \iint \left| \nabla \hat{a}_{\omega_{x,y}}^j \right|^2 d\omega_x d\omega_y \leq \max_{0 < \lambda < R} \frac{\frac{\pi}{2} (R^4 - (R - \lambda)^4)}{\pi (R^2 - (R - \lambda)^2) \eta_{mult}^2 + \eta_{add}^2}. \quad (21)$$

For  $J$  disjoint codes, the best gradient power is obtained when their union forms an outer ring and Eq. (16) follows.  $\square$

**Corollary 1.** *The total depth discrimination score, oriented derivative power and gradient power are bounded in the following order*

$$\mathfrak{D}(\hat{\mathbf{a}}) \leq \mathfrak{C}(\hat{\mathbf{a}}) \leq \mathfrak{G}(\hat{\mathbf{a}}) \leq \mathfrak{B}. \quad (22)$$

*Optimal discrimination codes should therefore satisfy the following three properties:*

1. *The total gradient power should approach the bound  $\mathfrak{G}(\hat{\mathbf{a}}) \rightarrow \mathfrak{B}$ .*
2. *For every  $\omega_x, \omega_y$ , the oriented derivative power should approach the gradient power  $\mathfrak{C}_{\omega_x, y}(\hat{\mathbf{a}}) \rightarrow \mathfrak{G}_{\omega_x, y}(\hat{\mathbf{a}})$*
3. *For every  $\omega_x, \omega_y$ , the discrimination score should approach the oriented derivative power  $\mathfrak{D}_{\omega_x, y}(\hat{\mathbf{a}}) \rightarrow \mathfrak{C}_{\omega_x, y}(\hat{\mathbf{a}})$*

To understand the first property note that the proof of Claim  $\square$  implies that the gradient power is maximized when the codes let in light at a ring at the periphery of the aperture. The exact ring width is a function of the ratio between the multiplicative and additive noise components. If the noise is fully additive ( $\eta_{mult} = 0$ ) it is best to collect light over the entire aperture area. When the noise is mostly multiplicative ( $\eta^2 \approx \alpha \eta_{mult}^2$ ), the best is to have a very narrow ring at the periphery of the aperture.

The oriented derivative power is equal to the gradient power if and only if for all  $j$ s, the gradient direction at spatial frequency  $\omega_{x,y}$  equals  $\omega_{x,y}$ . Having all gradients oriented toward the center implies that  $\hat{a}^j$  should be radially symmetric.

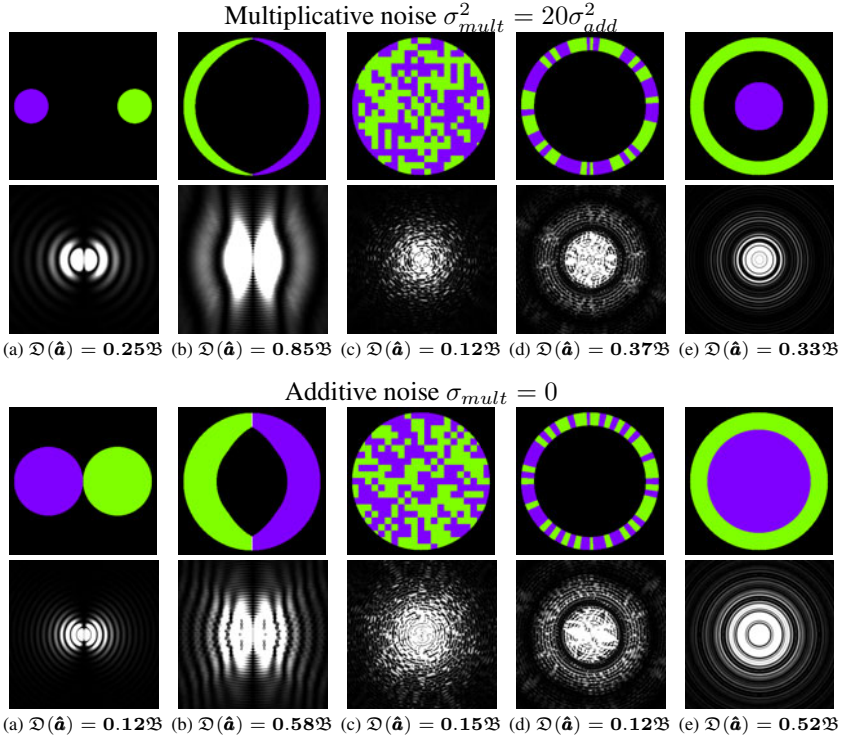
The last property of Corollary  $\square$  requires that the OTFs vector is orthogonal to its derivative. This property is the hardest to analyze, but in the next section we consider examples of codes which approach this requirement.

## 4 Analyzing Aperture Code Designs

In this section we consider a few aperture code designs and analyze their optimality. We start with designs capturing two images and then move to multiple images.

### 4.1 Two Image Designs

We consider the aperture code pairs visualized in Figure  $\square$ . The first one is a standard stereo setting with two disc holes. Another design is two halves of a ring (bananas),



**Fig. 1.** Aperture codes and their discrimination score. Top group: multiplicative noise ( $\sigma_{mult}^2 = 20\sigma_{add}^2$ ), Lower group: additive noise ( $\sigma_{mult}^2 = 0$ ). Each group visualizes in the upper row aperture codes (primal domain) and in the second row their discrimination score  $\mathfrak{D}_{\omega_{x,y}}(\hat{\mathbf{a}})$  provided in each spatial frequency (frequency domain). The portion of the upper bound utilized by  $\mathfrak{D}(\hat{\mathbf{a}})$  is reported at the bottom.

inspired by the optimized codes of [11]. We then consider pseudo random codes in each view. We consider random codes in the entire aperture area (Fig 1(c)) and only in the periphery ring (Fig 1(d)). Finally we consider a depth from defocus like pair (Fig 1(e))-an outside ring plus an inner disc.

We consider two noise situations, well illuminated scenes at which  $\eta_{mult}^2 = 20\eta_{add}^2$ , and the purely additive noise case  $\eta_{mult} = 0$ . Apart for the random designs, we searched numerically for good parameters in each pair family (e.g. the width of the rings and discs, or the exact shape of the banana parameterized as a spline with 5 keypoints). The parameters optimization is done independently for each noise situation. For multiplicative noise, narrower holes or rings at the periphery are favored and at the additive noise case wide code areas are selected.

For each pair, the second row of Fig 1 visualizes the map of discrimination score at each spatial frequency. At the bottom we report the portion of the bound achieved by the total discrimination scores. The best discrimination is obtained by the two halved rings, which at the multiplicative case utilize 85% of the gradient power upper bound.

All other designs utilize at most 50% of the bound. Below we discuss some of the interesting properties of each design.

Note that in designs (a-d) the codes are flipped versions of each other  $a_{(x,y)}^1 = a_{(-x,-y)}^2$ , and hence the spectra are conjugate. To analyze the discrimination scores of code pairs of this type we derive the following lemmas. The first lemma shows that the spectrum derivative consists of the magnitude derivative plus the phase derivative, and for conjugate spectra the portion of the derivative lost in the non orthogonal direction is the derivative of the magnitude. Using this observation we will later aim to show that for discriminative code pairs the magnitude derivative is small relative to the phase.

**Lemma 1.** *Let  $\hat{a}^1, \hat{a}^2$  be a conjugate pair of aperture spectra,*

$$\hat{a}_{\omega_{x,y}}^1 = m_{\omega_{x,y}} e^{i\zeta_{\omega_{x,y}}}, \quad \hat{a}_{\omega_{x,y}}^2 = m_{\omega_{x,y}} e^{-i\zeta_{\omega_{x,y}}}. \quad (23)$$

where  $m_{\omega_{x,y}}$  is the magnitude and  $\zeta_{\omega_{x,y}}$  the phase. The discrimination score equals the magnitude times the phase derivative power which is also the derivative power minus the magnitude derivative power:

$$\mathfrak{D}_{\omega_{x,y}}(\hat{\mathbf{a}}) = 2 |m_{\omega_{x,y}}|^2 |\nabla_{\omega_{x,y}} \zeta_{\omega_{x,y}}|^2 = |\nabla_{\omega_{x,y}} \hat{\mathbf{a}}|^2 - 2 |\nabla_{\omega_{x,y}} |\hat{\mathbf{a}}||^2. \quad (24)$$

*Proof.* The derivatives are the sum of the magnitude derivative and the phase derivative:

$$\nabla_{\omega_{x,y}} \hat{a}_{\omega_{x,y}}^1 = e^{i\zeta_{\omega_{x,y}}} \nabla_{\omega_{x,y}} m_{\omega_{x,y}} + m_{\omega_{x,y}} e^{i\zeta_{\omega_{x,y}}} i \nabla_{\omega_{x,y}} \zeta_{\omega_{x,y}} \quad (25)$$

$$\nabla_{\omega_{x,y}} \hat{a}_{\omega_{x,y}}^2 = e^{-i\zeta_{\omega_{x,y}}} \nabla_{\omega_{x,y}} m_{\omega_{x,y}} - m_{\omega_{x,y}} e^{-i\zeta_{\omega_{x,y}}} i \nabla_{\omega_{x,y}} \zeta_{\omega_{x,y}}. \quad (26)$$

Since the two phase derivatives have opposite signs they cancel each other when we take the inner product with  $\hat{\mathbf{a}}$  and we are left with the magnitude derivative only:

$$\frac{|\hat{\mathbf{a}}_{\omega_{x,y}}^* \nabla_{\omega_{x,y}} \hat{\mathbf{a}}_{\omega_{x,y}}|^2}{|\hat{\mathbf{a}}_{\omega_{x,y}}|^2} = 2 |\nabla_{\omega_{x,y}} m_{\omega_{x,y}}|^2 \quad (27)$$

and Eq. (24) follows from the definition in Eq. (12).  $\square$

Next, we note that to analyze the spectra  $\hat{\mathbf{a}}$  along direction  $\theta$  we can apply the slicing theorem [18] and look at the projection of the primal codes  $\mathbf{a}$  in the orthogonal direction. Let  $\rho^{j,\theta}$  denote the projection of  $a^j$  onto direction  $\theta$

$$\rho_{(x)}^{j,\theta} = \int a_{(x \vec{\theta} + y \vec{\theta}^+)}^j dy, \quad (28)$$

for  $\vec{\theta} = (\cos(\theta), \sin(\theta))$ ,  $\vec{\theta}^+ = (-\sin(\theta), \cos(\theta))$ . Let  $|\rho^{j,\theta}|^2 = \int |\rho_{(x)}^{j,\theta}|^2 dx$  denote the total power and  $\tau^{j,\theta}$  the power center of mass:

$$\tau^{j,\theta} = \frac{1}{|\rho^{j,\theta}|^2} \int |\rho_{(x)}^{j,\theta}|^2 x dx. \quad (29)$$

If  $a^1, a^2$  are a flipped pair, their powers are equal and their mass centers satisfies  $\tau^{1,\theta} = -\tau^{2,\theta}$ . Therefore we can think of  $2|\tau^{j,\theta}|$  as the average disparity along direction  $\theta$ .

The following lemma shows that if the projection has a relatively wide disparity, there will be a large discrimination in that direction. Figure 2 visualizes projections of the codes in Figure 1(a-d). As we will explain below, the more discriminative codes have a wider disparity in most orientations.

**Lemma 2.** Let  $a^1, a^2$  be a pair of flipped aperture codes. The frequency domain discrimination score along direction  $\theta$  is at least the square of the averaged disparity times the total power

$$\int \mathfrak{D}_{\omega \vec{\theta}}(\hat{\mathbf{a}}) d\omega \geq \frac{2}{\eta^2} |\tau^{j,\theta}|^2 |\rho^{j,\theta}|^2 \quad (30)$$

*Proof.* The slicing theorem [18] implies that a slice of  $\hat{a}^j$  in direction  $\vec{\theta}$  is the 1D Fourier transform of  $\rho^{j,\theta}$ . Therefore, we can apply a 1D variant of Claim 1 to compute the total derivative magnitude of  $\hat{a}^j$  along direction  $\vec{\theta}$ .

$$\int \left| \nabla_{\vec{\theta}} \hat{a}^j(\omega \vec{\theta}) \right|^2 d\omega = \int (x \rho_{(x)}^{j,\theta})^2 dx. \quad (31)$$

We note that

$$\int (x \rho_{(x)}^{j,\theta})^2 dx = \int (x - \tau^{j,\theta})^2 (\rho_{(x)}^{j,\theta})^2 dx + |\tau^{j,\theta}|^2 |\rho^{j,\theta}|^2. \quad (32)$$

Our goal is to show that

$$\int \left| \nabla_{\vec{\theta}} m_{\omega \vec{\theta}} \right|^2 d\omega \leq \int (x - \tau^{j,\theta})^2 |\rho_{(x)}^{j,\theta}|^2 dx, \quad (33)$$

and then Eq. (30) will follow directly from Lemma 1 (the factor 2 is because we sum over 2 code spectra). For that note that for any phase  $\chi_{\omega \vec{\theta}}$ ,

$$\left| \nabla_{\vec{\theta}} \left( m_{\omega \vec{\theta}} e^{i\chi_{\omega \vec{\theta}}} \right) \right|^2 = \left| \nabla_{\vec{\theta}} m_{\omega \vec{\theta}} \right|^2 + \left| m_{\omega \vec{\theta}} \right|^2 \left| \nabla_{\vec{\theta}} \chi_{\omega \vec{\theta}} \right|^2 \geq \left| \nabla_{\vec{\theta}} m_{\omega \vec{\theta}} \right|^2. \quad (34)$$

In particular, we can choose a phase  $\chi$  such that  $m_{\omega \vec{\theta}} e^{i\chi_{\omega \vec{\theta}}}$  will be the Fourier transform of  $|\rho_{(x-\tau^{j,\theta})}^{j,\theta}|^2$  (that is, a centered version of  $\rho$ ). Applying Claim 1 again, the total derivative power in that centered version is

$$\int (x \rho_{(x-\tau^{j,\theta})}^{j,\theta})^2 dx = \int (x - \tau^{j,\theta})^2 (\rho_{(x)}^{j,\theta})^2 dx \quad (35)$$

and Eq. (33) follows.  $\square$

Another intuitive argument which follows from the above proof is that when most of the mass of  $\rho^{j,\theta}$  is located in a narrow region, but the average disparity is large, the magnitude derivative power is small relative to the total derivative power. This is because the total derivative power is  $\int (x \rho_{(x)}^{j,\theta})^2 dx$ , and the magnitude derivative power is bounded by  $\int (x \rho_{(x-\tau^{j,\theta})}^{j,\theta})^2 dx$ . The  $x^2$  values to which  $(\rho_{(x-\tau^{j,\theta})}^{j,\theta})^2$  assigns non-zero weight are small compared to the non-zero values of  $(\rho_{(x)}^{j,\theta})^2$ . Since Lemma 1 shows that the portion of the derivative power lost in the non-orthogonal direction is the magnitude derivative power, if the magnitude derivative power is low, most of the derivative power  $\mathfrak{C}(\hat{\mathbf{a}})$  contributes to the discrimination score  $\mathfrak{D}(\hat{\mathbf{a}})$ .

We now use Lemmas 1 and 2 to analyze the code pairs in Fig 1

**Stereo pair:** (Fig 1(a)). The main problem with this design is that it provides disparity only in the horizontal direction. The discrimination map in Fig 1(a) is high around the horizontal spatial frequency axis and low around the vertical one. Similarly, the projections in Fig 2(a) show that the average disparity is large in the vertical direction,

but reduces as the projection angle changes, with zero disparity horizontally. Stereo violates the second optimal discrimination property of Corollary 1. The spectra are not radially symmetric, but the gradient is mostly in the horizontal direction at all spatial frequencies. To see this, let  $m_{\omega_x, y}$  denote the spectrum of a disc centered at  $(0, 0)$ . The spectrum of a disc centered at a point  $\tau = [\tau_x, \tau_y]$  is a phase shifted version of  $m$ :

$$\hat{a}_{\omega_x, y}^1 = m_{\omega_x, y} e^{i(\tau_x \omega_x + \tau_y \omega_y)}, \quad \hat{a}_{\omega_x, y}^2 = m_{\omega_x, y} e^{-i(\tau_x \omega_x + \tau_y \omega_y)}. \quad (36)$$

According to Lemma 1, the magnitude derivative does not contribute to the discrimination, and the phase gradient has a constant direction  $\tau$ . In fact, this implies that the total discrimination score of a stereo pair cannot pass 50% of the total gradient power. To see this, note that the discrimination score at frequency  $\omega_{x, y}$  is a function of the inner product of  $\tau$  and the direction  $\omega_{x, y}/|\omega_{x, y}|$ :  $\mathfrak{D}_{\omega_{x, y}}(\hat{a}) = \frac{2|m_{\omega_{x, y}}|^2}{|\omega_{x, y}|^2} |\omega_{x, y}^* \tau|^2$  (the factor 2 results from summing  $J = 2$  spectra). Since  $m$  is radially symmetric, averaged over all directions, this inner product utilizes only half of the power of  $\tau$ . Hence:

$$\mathfrak{D}_{\omega_{x, y}}(\hat{a}) = \iint \frac{2|m_{\omega_{x, y}}|^2}{|\omega_{x, y}|^2} |\omega_{x, y}^* \tau|^2 d\omega_x d\omega_y = \iint |m_{\omega_{x, y}}|^2 |\tau|^2 d\omega_x d\omega_y \leq \frac{1}{2} \mathfrak{G}_{\omega_{x, y}}(\hat{a}). \quad (37)$$

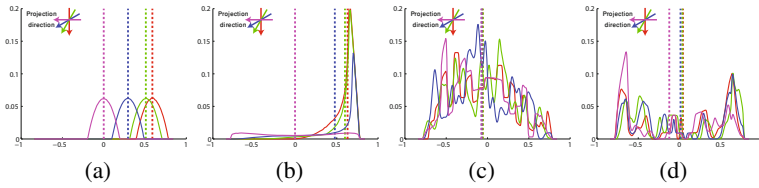
In Fig 1(a) we see that for multiplicative noise two smaller holes at the far ends of the aperture are preferred, while for additive noise it is better to have wide holes at the expense of reducing the disparity between them.

**Halved rings:** (Fig 1(b)). Our numerical calculation shows that for multiplicative noise this design utilizes 85% of the bound. There is no simple closed-form formula for these code spectra. However, we provide below a few intuitive arguments to justify this success. The important property of halved rings is a large disparity along most orientations. Fig 2(b) visualizes the 1D oriented code projections. Apart from the horizontal projection, the mean disparity at most other orientations is close to the aperture boundary, and hence Lemma 2 implies high discrimination at most orientations.

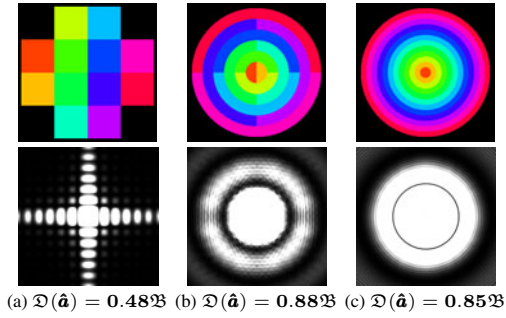
For this near-optimal pair we can verify empirically that all three properties of Corollary 1 hold. Property 1: the open code area is next to the periphery of the aperture disc. Property 2: for most orientations, the projections in Fig 2(b) are similar, and have the shape of a narrow peak at the right plus a tail (the exact tail shape varies between different projection directions). Since most projections are similar, most oriented slices from the spectra are similar, and the spectra are relatively radially symmetric except of a narrow angle range around the vertical direction. Property 3: according to the discussion after Lemma 2 the fact that the 1D projections have most mass around a narrow peak implies that the magnitude derivative power is small, which implies by Lemma 1 that most of the derivative power is discrimination power.

When the noise is additive the codes area is wider and the optimal shape resemble the one discovered by Zhou et al. [11] (who assume an additive noise). The banana's shape of the ends reduces the degeneracy around the vertical direction.

**Pseudo random codes:** (Fig 1(c,d)) obtain the worst discrimination scores. In terms of Lemma 2, the fact that light is collected from all around the aperture means that there is no real disparity, and the mean projections in Fig 2(c,d) are close to zero. Also, these codes violate properties 2 and 3 of Corollary 1. They are not radially symmetric and



**Fig. 2.** 1D projections  $\rho_\theta$  for the first 4 codes of Figure 1 multiplicative noise. We plot projections at 4 sampled orientations  $0^\circ$  (red),  $30^\circ$  (green),  $60^\circ$  (blue) and  $90^\circ$  (magenta). For each projection direction the center of mass (mean disparity) is marked by a dashed line.



**Fig. 3.** Top row: Aperture codes which divide the aperture into multiple images. Second row: The discrimination  $\mathcal{D}_{\omega_{x,y}}(\hat{\mathbf{a}})$  they provide in each spatial frequency. At the bottom we provide the portion of the upper bound achieved by  $\mathcal{D}(\hat{\mathbf{a}})$  for the additive noise case.

the derivatives are not orthogonal to the OTFs. In the multiplicative noise case the full aperture code (Fig 1(c)) has another major problem since it lets in light over the entire aperture area and not only around the boundaries. This is solved by a code in the outer ring (Fig 1(d)), but properties 2,3 are still problematic.

**Ring and disc pair:** (Fig 1(e)). This design should simulate the concept of depth from defocus (DFD). Intuitively, its drawback is enabling only half of the possible disparity—from the aperture boundary to the center, instead of from end to end. While this design is perfectly radially symmetric and satisfies property 2 of Corollary 1 for multiplicative noise it violates property 1, as it lets in light at the center of the aperture and holes at the center of the aperture do not contribute to spectra gradients. This design also violates property 3 since the OTFs vector is not orthogonal to its derivative.

**Stereo v.s. depth from defocus:** Our analysis predicts that DFD overcomes stereo. The disadvantages of stereo are having disparity only along one axis, and in the additive noise case it also suffers because it blocks light. This is consistent with previously reported evaluations of stereo v.s. DFD under similar physical dimensions [11–13].

**4.2 Multiple Image Designs**

We now consider some designs which capture a large number of images. We note that while an image captured from an inner aperture area is less discriminative than an image

from an outer ring, adding additional images always improves the discrimination score. Therefore if we are allowed to capture a large number of images we do want to utilize the entire aperture area. If the images number is unlimited, better results can be obtained if we restrict the area of each code such that  $\alpha \rightarrow 0$  and the noise reduces  $\eta^2 \rightarrow \eta_{add}^2$ .

The first example divides the aperture into multiple subsquares (Fig 3(a)). It is sub-optimal and must lose at least 50% of the gradient power budget, since as in the stereo case the gradients have a constant direction. This design is similar to the plenoptic camera implementation of [5] which divides the aperture area and captures multiple views ([5] divides the sensor area as well).

In contrast, Fig 3(b,c) show two near-optimal designs utilizing over 85% of the bound. The first one generalizes Fig 1(b) with a set of half rings. To remove the degeneracy in one direction, it alternates vertical and horizontal pairs. The second near-optimal design is a set of annular rings, like the multi-aperture camera of [2]. Despite the sub-optimality of the ring and disc in Fig 1(e), multiple rings can approach the bound.

## 5 Discussion

In this paper we have analyzed the depth discrimination accuracy provided by a general set of aperture codes. We propose an analytic upper bound on the best achievable discrimination, and study the desired characteristics of an optimal solution. We show that under multiplicative noise, the two half-ring codes of [1] provide near-optimal discrimination. When a large number of images are allowed, a multi-aperture camera [2] dividing the aperture into annular rings provides near-optimal discrimination. In contrast, a plenoptic camera can achieve at most 50% of the bound.

Our analysis and bounds can be extended to more general families of computational cameras such as cameras including phase plates (optical elements with non standard curvature) and not only amplitude masks. This, however, requires an analysis of the lens kernels in the 4D light field spectrum space, as proposed in [10].

## References

1. Zhou, C., Lin, S., Nayar, S.K.: Coded Aperture Pairs for Depth from Defocus. In: ICCV (2009)
2. Green, P., Sun, W., Matusik, W., Durand, F.: Multi-aperture photography. In: SIGGRAPH (2007)
3. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: IJCV (2002)
4. Adelson, E., Wang, J.: Single lens stereo with a plenoptic camera. In: IEEE PAMI (1992)
5. Georgeiv, T., Zheng, K., Curless, B., Salesin, D., Nayar, S., Intwala, C.: Spatio-angular resolution tradeoffs in integral photography. In: EGSR (2006)
6. Chaudhuri, S., Rajagopalan, A.: Depth from defocus: A real aperture imaging approach. Springer, New York (1999)
7. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: Mask-enhanced cameras for heterodyned light fields and coded aperture refocusing. In: SIGGRAPH (2007)

8. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. In: SIGGRAPH (2007)
9. Dowski, E., Cathey, W.: Single-lens single-image incoherent passive-ranging systems. *App. Opt.* (1994)
10. Levin, A., Hasinoff, S., Green, P., Durand, F., Freeman, W.: 4D frequency analysis of computational cameras for depth of field extension. In: SIGGRAPH (2009)
11. Schechner, Y., Kiryati, N.: Depth from defocus vs. stereo: How different really are they. In: *IJCV* (2000)
12. Farid, H., Simoncelli, E.: Range estimation by optical differentiation. In: *JOSA* (1998)
13. Vaish, V., Levoy, M., Szeliski, R., Zitnick, C., Kang, S.: Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In: *CVPR* (2006)
14. Levin, A., Freeman, W., Durand, F.: Understanding camera trade-offs through a Bayesian analysis of light field projections. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV. LNCS*, vol. 5305, pp. 88–101. Springer, Heidelberg (2008)
15. Hasinoff, S., Kutulakos, K.: Light-efficient photography. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV. LNCS*, vol. 5305, pp. 45–59. Springer, Heidelberg (2008)
16. Hasinoff, S., Kutulakos, K., Durand, F., Freeman, W.: Time-constrained photography. In: *ICCV* (2009)
17. Zhou, C., Nayar, S.K.: What are Good Apertures for Defocus Deblurring? In: *IEEE International Conference on Computational Photography* (2009)
18. Ng, R.: Fourier slice photography. In: SIGGRAPH (2005)



# We Are Family: Joint Pose Estimation of Multiple Persons

Marcin Eichner and Vittorio Ferrari

Computer Vision Laboratory, ETH Zurich, Switzerland  
{eichner, ferrari}@vision.ee.ethz.ch

**Abstract.** We present a novel multi-person pose estimation framework, which extends pictorial structures (PS) to explicitly model interactions between people and to estimate their poses jointly. Interactions are modeled as occlusions between people. First, we propose an occlusion probability predictor, based on the location of persons automatically detected in the image, and incorporate the predictions as occlusion priors into our multi-person PS model. Moreover, our model includes an inter-people exclusion penalty, preventing body parts from different people from occupying the same image region. Thanks to these elements, our model has a global view of the scene, resulting in better pose estimates in group photos, where several persons stand nearby and occlude each other. In a comprehensive evaluation on a new, challenging group photo datasets we demonstrate the benefits of our multi-person model over a state-of-the-art single-person pose estimator which treats each person independently.

## 1 Introduction

Look at the photo in Figure [1a](#). A group of people poses for a souvenir picture. The majority of body parts of the people in the back row are occluded by the people in the front row. Many photos of this kind can be found in personal photo collections or on community sites like Flickr or Facebook.

Unfortunately, even state-of-the-art 2D articulated human pose estimation (HPE) algorithms [\[1-3\]](#) typically fail on such photos (Fig. [1b](#)). These failures are due to treating every person independently, disregarding their interactions. As HPE algorithms lack a global view on the scene, they cannot handle occlusions between people.

In this paper we propose a pose estimation approach which explicitly models interactions between people and estimates their poses *jointly*. Our model handles occlusions between people and prevents body parts of neighboring persons from covering the same image region (Fig. [1c](#)).

Our main contributions are: (i) an algorithm to predict the probability that a body part of a person is occluded given only the locations of all persons in the image (without knowing their poses); (ii) a novel model extending pictorial structures to jointly estimate the pose of multiple persons. This model incorporates the occlusion predictor as well as mutual exclusion terms preventing body parts from different people in the same image region. We also give an efficient inference technique for this model; (iii) a new dataset of group photos fully annotated with a labeling of which body parts are visible/occluded and with the location of visible parts.



**Fig. 1. Group photo scenario.** (a) example image; (b) result of independent pose estimation [1]; (c) result of our joint multi-person pose estimation.

We demonstrate experimentally on the new group photo dataset that (i) the occlusion predictor performs well and better than various baselines, including an occlusion prior probability estimated from a training set; (ii) the whole joint multi-person algorithm considerably outperforms a state-of-the-art single-person estimator [1]. Our source code is available at [4].

**Related Works.** In this work we explore interactions between people to improve HPE in group photos. In this section we briefly review recent works on relevant topics.

Recovering articulated body poses is a challenging task. We build on Pictorial Structures [5], a popular paradigm for single-person HPE in still images [2, 3, 5, 6] (sec. 3.1).

As a part of our multi-person model we look at occlusions. In articulated HPE some previous works model self-occlusions [7–10]. Here instead we consider occlusions between people. Modeling interactions between people is at the core of our work. They were exploited before by multi-person trackers in the tracking-by-detection paradigm [11–14] (e.g. in [13] as space-time constraints preventing multiple people from occupying the same 3D space at the same time). In [14] the authors learn the behavior of people in crowded urban scenes and predict the path of a pedestrian given the location and direction of others. All these trackers [11–14] handle occlusions at the level of entire persons, who are considered as atomic units (and not at the level of body parts).

To the best of our knowledge, we are the first to propose a joint multi-person occlusion-sensitive model for articulated HPE, where interactions between people are modeled at the level of body parts.

## 2 We Are Family - Scenario and Dataset

A typical group photo contains several people standing nearby and occluding each others’ body parts. We argue that for such photos a joint multi-person reasoning is beneficial over estimating the pose of each person independently.

To investigate this claim we collected a new dataset of group photos, e.g. classmates, sport teams and music bands. We collected the images from Google-Images and Flickr using queries composed of words like “group”, “team”, “people”, “band” and “family”. The resulting dataset has 525 images with 6 people each on average. People appear upright in near-frontal poses and often occlude one another (Fig. 1a). They sometimes are even lined up in a few rows, which results in many occlusions. Across different images people appear at a variety of scales and illumination conditions.

The six upper-body parts have been annotated (head, torso, upper and lower arms). For each person, the annotation includes an occlusion vector  $\mathcal{H}$  indicating which body parts are visible/occluded, and a line segment for each visible body part. Moreover, the

depth order of the people in each image is also annotated, so we know who is in front of who. We plan to release this new dataset freely on-line.

### 3 Multi-person Pictorial Structure Model (MPS)

We first introduce the pictorial structure (PS) framework [5] for HPE of single persons (sec. 3.1), then we describe a naive extension to multiple persons and discuss its shortcomings (sec. 3.2) and finally we sketch our novel joint multi-person method (sec. 3.3).

#### 3.1 Single-Person Pictorial Structures (1PS)

**PS Model.** In the PS framework [5], a person’s body parts are nodes tied together in a Conditional Random Field [15]. Parts  $l_i$  are rectangular image patches and their position is parametrized by location  $(x, y)$ , orientation  $\theta$ , scale  $s$ , and sometimes fore-shortening [5, 16]. This parametrization constitutes the state-space of the nodes in the PS. The posterior of a configuration of parts  $L = \{l_i\}$  given an image  $I$  is

$$P(L | I, \Theta) \propto \exp \left( \sum_{(i,j) \in E} \Psi(l_i, l_j) + \sum_i \Phi(l_i | I, \Theta) \right) \quad (1)$$

The pairwise potential  $\Psi(l_i, l_j)$  is a prior on the relative position of two parts. It embeds kinematic constraints (e.g. the upper arms must be attached to the torso) and, in a few works, also other relations such as self-occlusion constraints [7] or the coordination between parts [17] (e.g. the balance between arms and legs during walking).

In many works the model structure  $E$  is a tree [2, 3, 5, 6], which enables exact inference, though some works explored more complex topologies [2, 7, 16, 17].

The unary potential  $\Phi(l_i | I, \Theta)$  corresponds to the local image evidence for a part in a particular position (likelihood) and it depends on appearance models  $\Theta$  describing how parts look like.

**Appearance Models.** The success of PS depends critically on having good appearance models  $\Theta$ , which limit the image positions likely to contain a part. In an effort to operate on a single image, with unknown part appearances, Ramanan [6] proposes *image parsing*. In this approach  $\Theta$  are improved iteratively, by adding person specific appearance models computed from previously estimated pose, where the first pose is obtained using only generic edge models as unary potentials.

**Person Detection.** As in several recent HPE methods [1, 12, 13], we use a generic person detector to determine the approximate location and scale of the persons in an image. This was shown to be useful for estimating pose in uncontrolled, cluttered images [2], as it reduces the state-space of the PS nodes by fixing the scale and reducing the  $(x, y)$  search region to an enlarged area around the detection window. In this paper, the set of detection windows  $\mathcal{D}$  also determines the set of people  $\mathcal{P}$  in the image.

In [1] authors also use the initial detection to obtain person-specific appearance models  $\Theta$  from a single image (as an alternative to [6]). They propose to compute  $\Theta$  using part specific *segmentation priors*, learned wrt  $\mathcal{D}$ , and then improve  $\Theta$  using an *appearance transfer* mechanism that exploits between part appearance dependencies.

### 3.2 Naive Multi-person Algorithm

Given an image with a set of people  $\mathcal{P}$ , a simple algorithm for multi-person HPE could be: (1) estimate the pose of the first person using an off-the-shelf algorithm, e.g. [1, 3]; (2) remove the support of the estimated pose from the image, e.g. erase pixels covered by the pose; (3) repeat (1)-(2) for the next person.

We call *front-to-back order (FtB)*  $Z$  the sequence in which people are processed by the algorithm. Since the true depth ordering is unknown, one must run the algorithm for  $|\mathcal{P}|!$  different orders and then pick the best one according to some criterion, e.g. the product of eq. (1) over people.

There are three problems with the naive algorithm: (i) it is infeasible to run due to the factorial complexity in the number of people  $|\mathcal{P}|$ ; (ii) such a greedy algorithm doesn't have a global view on the scene, so people interactions are limited to removing image evidence; (iii) typical HPE algorithms [1, 3] don't handle occlusions and always try to find an image position for all body parts (except [7] for self-occlusions). Therefore, even if the naive algorithm ran over all the orders, it would not find out which parts are occluded. Removing image evidence in step (2) might even lead to double-counting, e.g. when both arms of a person are assigned to the same image region.

### 3.3 Our Approach to Multi-person Pose Estimation (MPS)

We propose a joint multi-person Pictorial Structures model which explicitly takes complex interactions between people into account:

$$P(\mathcal{L} | I, \theta, Z) \propto \exp \left( \sum_{p \in \mathcal{P}} \sum_{(i,j) \in E} \Psi(l_i^p, l_j^p) + \sum_{p \in \mathcal{P}} \sum_i \Phi(l_i^p | I, \theta, Z) + \sum_{(p,q) \in \mathcal{X}} \sum_i \sum_j a_{ij} \omega(l_i^p, l_j^q) \right) \quad (2)$$

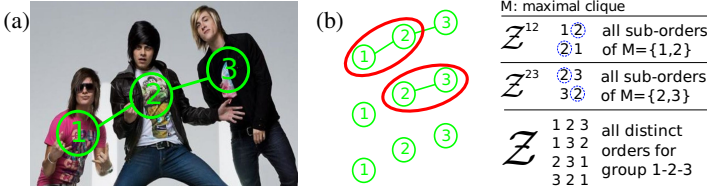
where the first term is a kinematic constraint as in the 1PS model (eq. (1)), but with additional summations over people  $p \in \mathcal{P}$ .

Interactions between people are at the core of our work and play an important role in two terms of our joint model. First, the unary potential  $\Phi$  is updated to depend on the FtB order  $Z$  and to include an occlusion state (sec. 5.3). The probability of the occlusion state is estimated specific to each person and body part before running PS inference, as a function of the location of all other persons in the image as given by the detection windows  $\mathcal{D}$ . Moreover, in sec. 4 we propose techniques to strongly reduce the number of FtB orders that the algorithm has to try out, also based on the spatial distribution of detection windows. The second point where people interactions are modeled is the new inter-people exclusion term  $\omega$ , which prohibits body parts from different persons  $(p, q)$  to occupy the same region (sec. 7),  $\mathcal{X}$  is the set of interacting people (sec. 4).

In section 6 we show how to perform efficient approximate inference in our MPS model. Finally, sec. 9 presents a quantitative evaluation of the occlusion predictor and a comparison of our joint MPS model to 1PS.

## 4 Reducing the Number of Front-to-Back Orders

We propose exact and approximate ways to reduce the number of FtB orders  $\mathcal{Z}$ .



**Fig. 2. FtB Orders.** (a) Group example (b) Calculating distinct FtB orders, red: maximal cliques found, blue: position of the common node in each FtB order

### 4.1 Exact Reductions

A person can influence another person only if they are within a certain proximity. We define two persons to be *interacting* if they are closer than their arm extents (more precisely, if their enlarged detection windows overlap (Fig. 2a)). An image can then be represented as an interaction graph  $\mathcal{X}$  with nodes corresponding to people and edges corresponding to interactions (Fig. 2a).

**Group Independence.** We define *groups of people* as connected components  $\mathbb{G}$  in the interaction graph  $\mathcal{X}$ . The first reduction is found by observing that any two persons from different groups cannot interact (i.e. groups are independent). Hence, pose estimation can be run on each group independently, and so the effective total number of orders is reduced from  $|\mathcal{P}|!$  to  $\sum_{\mathbb{G} \in \mathbb{G}} |\mathbb{G}|!$

**Order Equivalence.** Within a group, different FtB orders might lead to the same pose estimation result. This is the case for orders 132 and 312 in the graph 1-2-3, as there is no direct interaction between nodes 1 and 3 (Fig. 2a). We say that the two orders are *equivalent*. If person 2 is in the back then the order between 1 and 3 has no influence on pose estimation results. Analogously, orders 213 & 231 are also equivalent. Hence, there are only 4 distinct FtB orders instead of  $3! = 6$  in the graph 1-2-3.

This intuition is formalized in the following algorithm to find all distinct FtB orders in a group  $\mathcal{G}$ : (1) find the maximal clique  $M$  of  $\mathcal{G}$ ; (2) keep a record which nodes are in  $M$  and then remove all intra-clique edges from  $\mathcal{G}$ ; (3) compute sub-orders of the maximal clique  $M$  as all permutations of its nodes ( $\mathcal{Z}^{12}$  and  $\mathcal{Z}^{23}$  in Fig. 2b); (4) repeat until there are no edges in  $\mathcal{G}$ ; (5) compute the distinct orders of  $\mathcal{G}$  as all permutations between sub-orders of all maximal cliques  $\mathcal{M}_{\mathcal{G}}$  found, concatenated at the position of the common node ( $\mathcal{Z}$  in Fig. 2b).

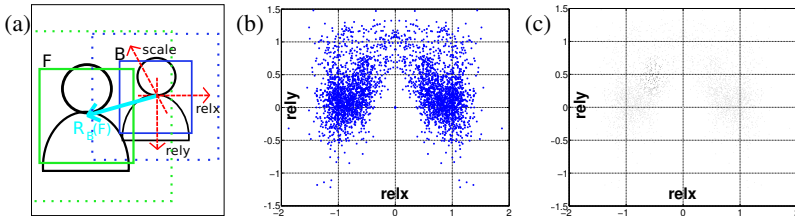
Group independence and order equivalence reduce the number of orders to:

$$\sum_{\mathbb{G} \in \mathbb{G}} \prod_{M \in \mathcal{M}_{\mathbb{G}}} |M|! \tag{3}$$

where  $\mathcal{M}_{\mathbb{G}}$  is the set of maximal cliques found in group  $\mathbb{G}$ .

### 4.2 Approximate Reductions

As the vast majority of group photos are taken parallel to the ground plane, the people appearing higher in the image are typically further away from the camera. Moreover if



**Fig. 3. Inter People Occlusion Probability.** (a): detection windows (solid), enlarged detection windows (dashed), (b) Distribution of relative locations of occluders  $f$  wrt the occluded person  $b$  over the training set. (c) relative contribution of the training points for test point  $[-0.5, 0.5, 1]$

a person appears larger than another, then it is likely closer to the camera. We propose two heuristics based on the spatial arrangement of the detected persons  $\mathcal{P}$  that capture these observations. Both estimate which person in a pair  $(p, q)$  is in front of the other:

**Relative Size.** If  $p$  is more than 2.5 times bigger than  $q$ , then  $p$  is in front of  $q$ .

**Relative Position.** If the center of  $p$  is higher than the top of  $q$ , then  $p$  is behind  $q$ .

## 5 Occlusion Probability (OP)

The visibility of the body parts of a person depends on her position with respect to other persons and wrt to the image border. We take both aspects into account by defining two types of occlusion probabilities. One type defines the probability that a part of a person is occluded, given the locations of all other persons and their FtB order (sec. 5.1). The other type defines the probability that a part is not visible, given the location of the person wrt to the image borders (sec. 5.2). We combine both probabilities specific to each person  $p$  and body part  $i$  into a single occlusion prediction  $\mathcal{O}_i^p$  (details in sec. 6), which is then used to set the energy of a new occlusion state in our MPS model (sec. 3.3).

### 5.1 Inter People Occlusion Probability (POP)

When two people are standing nearby it is likely that one is occluding some body parts of the other. The inter people occlusion probability (POP) is defined between a pair of interacting persons. One person  $f$  is considered to be in the front (occluder) and the other  $b$  in the back (according to the FtB order). POP tells how likely a body part  $l$  of  $b$  is occluded given the relative location  $\mathcal{R}_b(f)$  of  $f$  in  $b$ 's coordinate frame (i.e.  $\mathcal{R}_b(f) = [(x_b - x_f)/w_b, (y_b - y_f)/h_b, h_f/h_b]$ , with  $x, y, w, h$  the center, width, and height of a window (Fig. 3a).

**Learning.** We model POP as a non-parametric distribution  $P(l_i^b = o \mid f, \mathcal{T})$  where  $l_i^b$  is part  $i$  of the back person and  $\mathcal{T}$  is a set of training person pairs. For each pair  $(f, b)$ , the training data is  $\mathcal{R}_b(f)$ , defined as above, and the ground-truth occlusion vector  $\mathcal{H}^b$  of the back person  $b$  (which is annotated in our dataset (sec. 2)) (Fig. 3b).

To take into account the uncertainty of the detector around the true position of a person, we run it on the training images and then associate detection windows to the annotated persons (as in [1]). This gives the person windows used for training.

Every pair of interacting persons  $(f, b)$  leads to a training sample  $(\mathcal{R}_b(f), \mathcal{H}^b)$  (two persons interact if their enlarged windows overlap, sec. 4.1). We determine which is  $f$  using the true FtB order, which is annotated in our dataset (sec. 2).

**Test Time.** At test time, we compute the probability that a body part  $i$  of a new person  $p$  is occluded by a person  $q$  in front of her:

$$P(l_i^p = o | q, \mathcal{T}) = \sum_{(f,b) \in \mathcal{T}} \alpha^{qpf^b} \gamma_i^b \quad \text{with} \quad \alpha^{qpf^b} = \frac{\mathcal{N}(\|\mathcal{R}_p(q) - \mathcal{R}_b(f)\| | 0, \sigma)}{\sum_{(d,c) \in \mathcal{T}} \mathcal{N}(\|\mathcal{R}_p(q) - \mathcal{R}_c(d)\| | 0, \sigma)} \quad (4)$$

The weights  $\alpha^{qpf^b}$  are set according to normalized Gaussian-weighted Euclidean distances between the relative location of the test pair  $(q, p)$  and those of the training pairs  $\mathcal{T}$  (Fig. 3c). The resulting POP value is always in  $[0, 1]$ .

For a given FtB order  $Z$ , if person  $p$  is behind more than one occluder then the POP probability of her part  $i$  is:

$$P(l_i^p = o | Z, \mathcal{T}) = \max_{f \in \mathcal{F}_Z^p} P(l_i^p = o | f, \mathcal{T}) \quad (5)$$

with  $\mathcal{F}_Z^p$  the set of occluders of  $p$  in FtB order  $Z$ .

**FtB orders for POP.** Only the *immediate* neighborhood  $\mathcal{V}^p$  of person  $p$  in the interaction graph has an influence of her POP values. Therefore, the FtB orders for calculating POP are node-specific (as opposed to the FtB orders for pose estimation, which are group-specific (sec. 4.1)). Since  $\mathcal{V}^p$  has a star-like topology, all its maximum cliques  $\mathcal{M}$  have size 2, so the number of FtB orders affecting POP values for person  $p$  is  $|\mathcal{Z}^p| = 2^{|\mathcal{V}^p|}$ , typically much smaller than the number of FtB orders in her group.

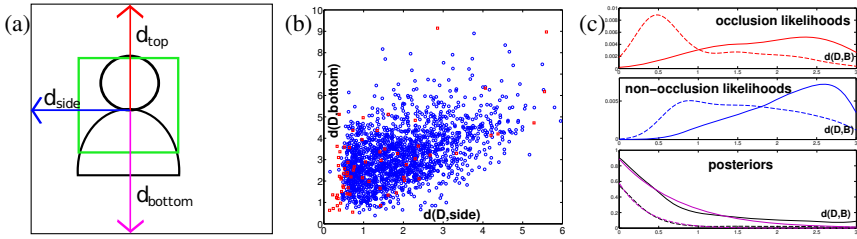
## 5.2 Border Occlusion Probability (BOP)

Some parts of a person might not be visible due to her proximity to an image border. We model here the probability of a part being occluded given the location of the person wrt to image borders  $\mathcal{B} = \{\text{top, bottom, side}\}$ .

We define BOP as  $P(l_i^p = o | d(D^p, B), \zeta_i^B)$  the probability that part  $i$  of person  $p$  is not visible given the normalized distance  $d(D^p, B)$  of her detection window  $D^p$  to a border  $B$  (Fig. 4a).  $\zeta_i^B$  are the parameters of the distribution.

**Learning.** To learn BOP we use our group photo dataset to collect training detection windows  $\mathcal{T}_D$  and associate them to ground-truth occlusion vectors  $\mathcal{T}_\mathcal{H}$  (as in sec. 5.1). For each type of body part  $i$  (e.g. right upper arm) and type of border  $B$ , we construct the occlusion  $P(d(D, B) | l_i = o)$  and non-occlusion  $P(d(D, B) | l_i \neq o)$  likelihoods as non-parametric kernel density estimates on the training data  $D \in \mathcal{T}_D$  (Fig. 4b). The Bayesian posterior of  $l_i$  being occluded given the distance to  $B$  is (Fig. 4):

$$P(l_i = o | d(D, B)) = \frac{P(d(D, B) | l_i = o)P(l_i = o)}{P(d(D, B) | l_i = o)P(l_i = o) + P(d(D, B) | l_i \neq o)(1 - P(l_i = o))} \quad (6)$$



**Fig. 4. Border occlusion probability.** (a) Distances to border types. (b) Distribution of  $d(D, B)$  wrt the bottom (y-axis) and side (x-axis) borders over the training set  $\mathcal{T}_D$ . Red dots: windows of persons with occluded upper arm. Blue dots: not occluded. (c) Top: example of occlusion likelihoods  $P(d(D, B) | l_i = o)$  for upper arms wrt to side and bottom borders (dashed and solid curves respectively). Middle: as top but for non-occlusion likelihoods  $P(d(D, B) | l_i \neq o)$ . Bottom: as top but for posterior distributions  $P(l_i = o | d(D, B))$  (in black) and their parametric approximations  $Y_\zeta(x)$  (in magenta) cropped to the range of the posteriors.

where  $P(l_i = o)$  is a prior calculated as the frequency of occlusion of part  $i$  over the training set. We approximate the non-parametric posterior estimates  $P(l_i = o | d(D, B))$  with a parametric function  $Y_\zeta(x) = c\mathcal{N}(x | \mu, \sigma)$ , fitted in the least square sense. This makes BOP more compact and does not restrict the image size at test time. As Fig. 4c shows, the fitted functions are very close to the non-parametric posteriors.

**Test Time.** At test time, we compute the probability that a body part  $i$  of a new person  $p$  is not visible wrt to each border type  $B \in \mathcal{B}$  and then select the maximum:

$$P(l_i^p = o | D^p, \mathcal{B}, \zeta_i^B) = \max_{B \in \mathcal{B}} P(l_i^p = o | d(D^p, B), \zeta_i^B) = \max_{B \in \mathcal{B}} Y_{\zeta_i^B}(d(D^p, B)) \quad (7)$$

where  $\zeta_i^B = \{c, \mu, \sigma\}_i^B$  are the parameters of the posterior approximation  $Y_\zeta(x)$  for border type  $B$  and part type  $i$ .

### 5.3 Incorporating Occlusion in the MPS Model

To handle occlusions in our MPS model (eq. (2)) we add an occlusion state to the state-space of the nodes (body parts). This results in an additional entry in the unary appearance likelihood  $\Phi(l_i | I, \Theta, Z)$  and an additional row and column in the pairwise kinematic prior  $\Psi(l_i, l_j)$ .

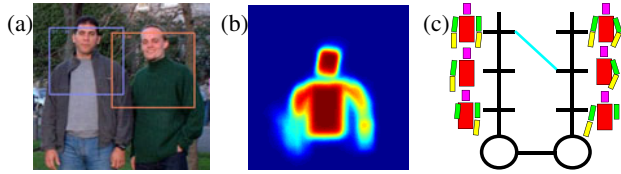
We consider the head as the root of the kinematic tree and set the pairwise term so that if a node is occluded, then all its children must be occluded as well. We consider the head to be always visible and give it no occlusion state.

We use the occlusion prediction  $\mathcal{O}_i^p$  to set the energy of the occlusion state in the extended MPS model (and the energies of the corresponding row/columns in the pairwise term). Therefore, the MPS model sees  $\mathcal{O}_i^p$  as a prior for a part to be occluded.

## 6 Inference

To find the optimal configuration of body parts in our joint MPS model (eq. (2)) we must minimize its energy also over FtB orders  $\mathcal{Z}$ . This is infeasible due to the factorial num-





**Fig. 5. Inference.** (a) an inference example, (b) a stack of samples drawn from the joint probability of configuration of the left person, (c) puppet state-space graphical model (eq. (8)), the lowest energy configuration according to the joint model is marked by the cyan line.

ber of orders in the number of persons and the relatively high cost of pose estimation for a person. The techniques we propose in sec. 4 bring us closer to the goal, as they greatly reduce the number of orders to be considered. Yet, it remains inconveniently expensive to find the exact global optimum. Therefore, we show here how to perform efficient approximate optimization of eq. (2) over  $\mathcal{L}$ . Notice that the optimization is done only once as all FtB orders are marginalized out while computing POP (sec. 5.1).

**Person-level model.** The key idea is to rewrite eq. (2) on a coarser level, where a node is a person rather than a body part:

$$P(\mathcal{L} | I, \Theta) \propto \exp \left( \sum_{u \in \mathcal{U}} \sum_{p \in \mathcal{P}} u(L^p | I, \Theta) + \sum_{(p,q) \in \mathcal{X}} \Omega(L^p, L^q) \right) \quad (8)$$

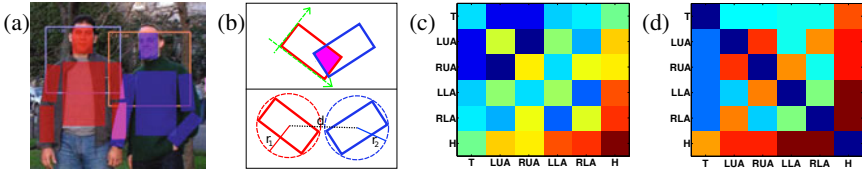
where  $\mathcal{U}$  is the set of unary terms related to one person and  $\Omega$  is the inter-person exclusion term (as  $\omega$  in eq. (2) but now defined on the person level). A single state for a node in eq. (2) was a particular location of a body part of a person, whereas in eq. (8) it is a spatial configuration of all body parts of a person - a puppet (Fig. 5c). All terms of eq. (2) relating to one person become unary terms  $u$  in eq. (8) (also the pairwise kinematic prior  $\Psi(l_i, l_j)$  between parts). The set of model edges corresponds to the interaction graph  $\mathcal{X}$ . The exclusion term  $\Omega$  is detailed in the next section, as it can be computed very efficiently by exploiting the properties of the puppet-space and of the inference algorithm below.

**Efficient inference.** This remodeling of the problem enables to take advantage of two important facts: (i) the occlusion probabilities POP/BOP depend on the output from the person detector only; (ii) the number of FtB orders that affects the occlusion probabilities is much smaller than the number of FtB orders affecting pose estimation (sec. 5.1). Based on these facts, we design the following approximate inference on the joint MPS model:

(1) *Compute the occlusion probability  $\mathcal{O}_i^p$  for every part of every person by combining POP (eq. (5)) and BOP (eq. (7)).* As at test time the FtB order is not given, we marginalize it out when computing POP (it does not affect BOP anyway):

$$\mathcal{O}_i^p = \max \left\{ P(l_i^p = o | D^p, \mathcal{B}, \zeta_i^{\mathcal{B}}), \frac{1}{|\mathcal{Z}^p|} \sum_{Z \in \mathcal{Z}^p} P(l_i^p = o | Z, \mathcal{T}) \right\} \quad (9)$$

(2) *Sample a small set of candidate puppets  $\mathcal{S}^p$  for every person.* We reduce the joint model to contain only the image likelihood  $\Phi$  and kinematic prior  $\Psi$  terms for one



**Fig. 6. (a)-(c) Exclusion between people.** (a) Blue and red overlays show likely body configuration for a single person HPE algorithm [11], magenta depicts image areas covered by both persons. (b) top: fast IoU between two rectangles using Sutherland-Hodgman clipping algorithm [20], bottom: bound on intersection of two rectangles, (c) limb-pairs contributions into the overall between people exclusion  $\Omega$ . **(d) Anti double-counting.** Limb-pair contributions (sec. 8)

person  $p$  and plug  $\mathcal{O}^p$  in the occlusion states as described in sec. 5.3. Then we sample 1000 puppets according to the posterior of the reduced model - a *proposal distribution* (Fig. 5b). When implemented efficiently, this sampling has computational complexity similar to finding the best puppet in the IPS model eq. (1).

(3) *Optimize the joint model.* We setup the state-space of each person  $p$  in the joint model (eq. (8)) to contain only the sampled puppets  $\mathcal{S}^p$  (Fig. 5c). As the interaction graph may contain loops, we use TRW-S [19] to run inference in this model. The computation time of this operation is negligible.

**Additional terms.** The final model eq. (8) contains additional unary terms in  $\mathcal{U}$ , defined on individual persons, designed to compensate flaws of the original PS formulation [5]. We explain them in more detail in sec. 8.

## 7 Exclusion between People $\Omega, \omega$

We explain here the inter-people exclusion term  $\Omega$ , which penalizes configurations where different people have body parts in the same image region (Fig. 6a).

We define it as  $\Omega(L^p, L^q) = \sum_i \sum_j a_{ij} \omega(l_i^p, l_j^q)$  where  $\omega(l_i^p, l_j^q)$  is the exclusion defined on per body part level and  $a_{ij}$  are per limb pair  $(i, j)$  weights (eq. (2));  $\omega(l_i^p, l_j^q)$  is defined as  $\log(1 - \text{IoU}(l_i^p, l_j^q))$  where  $\text{IoU}(l_i^p, l_j^q) \in [0, 1]$  is the area of intersection-over-union between body parts  $l_i^p, l_j^q$  of two persons  $p, q$  and a body part is approximated by a rectangle of constant aspect-ratio (Fig. 6a).

The inference approach of sec. 6 must compute the exclusion term between all pairs of body parts between all pairs of sampled puppets between all pairs of interacting people. If implemented directly this requires  $|\mathcal{S}|^2 * |L|^2 * |\mathcal{X}|$  IoU computations, where  $|\mathcal{S}|$  is the number of puppet samples,  $|L|$  the number of body parts, and  $|\mathcal{X}|$  the number of edges in the interaction graph  $\mathcal{X}$ . Although one IoU can be computed efficiently based on the Sutherland-Hodgman clipping algorithm [20] (Fig. 6b top), doing it for all pairs is very expensive.

We can drastically reduce the number of IoU computations without doing any approximation by observing that two rectangles  $i, j$  can have non-zero IoU only if the distance  $d(c_i, c_j)$  between their centers is smaller then the sum of the radii  $r_i, r_j$  of their circumscribing circles (Fig. 6b bottom). Therefore, we compute this bound for all pairs

of rectangles and then only compute IoU for the small minority with  $d(r_i, r_j) < r_i + r_j$ . The cost of computing the bound is negligible compared to the cost of IoU.

**Learning weights  $a_{ij}$ .** In our model (eq. (2)), the exclusion terms between different pairs of body parts  $(i, j)$  between two persons are combined in a sum weighted by  $a_{ij}$ . We learn these weights from the training set as follows. For each pair of parts  $(i, j)$ , we compute the average IoU  $m_{ij}$  between all pairs of interacting ground-truth stickmen (to avoid a bias, only pairs of not occluded parts contribute to the average). We then set  $a_{ij} = 1 - m_{ij}$ . As the arm of a person can be partially in front of her neighbor’s torso and yet both are visible, we want to penalize this situation little. Instead, we want to exclude that the arm of a person can overlap with the head of another. The learned weights follow these intuitions, and give a high weight for head-arm overlaps but lower weight to arm-torso overlaps (Fig. 6c).

## 8 Additional Single-Person Cues

We include in our joint model (eq. (8)) additional terms defined on individual persons, designed to compensate shortcomings of a plain PS model (sec. 3.1).

**Anti Double-Counting  $\Gamma$ .** The original PS formulation (eq. (1)) lacks a mechanism for discouraging two parts of the same person from explaining the same image region. This *double-counting* typically happens between left/right arms or legs. Several methods were proposed to tackle this problem including non-tree models [2, 17] and sequential image likelihood removal [16].

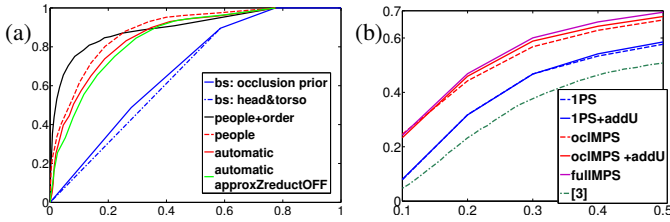
Interestingly, we can easily incorporate anti double-counting penalties in our model simply by adding a term analog to the inter-people exclusion  $\omega$ , but now between pairs of body parts of the same person. This is more principled than manually adding dependencies between parts incline to double-counting [2], as it enables to learn a weight for every part pair (in the same way as for  $\omega$ , sec. 7). The learned weights nicely lead to the desired behavior, e.g. high weights between all combinations of upper and lower arms, but low weights between arms and torso, resulting in a model that does not penalize configurations where the arms are in front of the torso (Fig. 6d).

**Foreground-Fill  $\lambda$ .** Foreground-fill  $\lambda$  encourages configurations of body parts colored differently than the background, similar to [21, 22]. It gives intermediate energies when body parts are occluded.

**Symmetric Arm Appearance  $\gamma$ .** Symmetric Arm Appearance  $\gamma$  encourages configurations where the left and right upper arms (as well as the left and right lower arms) have similar appearance. If one in a the pair is occluded, then it gives an intermediate energy to both.

## 9 Experiments and Conclusions

We present a comprehensive evaluation of (i) the algorithm’s complexity drop when using the FtB orders reductions (sec. 4); (ii) the ability of the method to predict which



**Fig. 7. Evaluation.** (a) ROC curves for binary occlusion classification (y-axis = true-positive rate, x-axis = false-positive rate). **Baselines:** *occlusion prior* - constant OP for each part set to the part’s frequency of occlusion over the training set; *head&torso* - head and torso always visible and all other parts always occluded. **Modes for our method:** *people+order* - ground-truth person detections and order  $Z$  given; *people* - only ground-truth detections given; *automatic* - true test scenario with nothing given; *automatic approxZreductOFF* - as *automatic* but without using the heuristics for reducing the number of FtB orders.

(b) PCP curves for pose estimation: *1PS* - [1]+[3], *1PS+addU* - [1]+[3] + additional single person terms  $\Gamma\Delta Y$  (sec. 8); *oclMPS* - the lowest energy puppet sampled from the *proposal distribution* of our MPS model including occlusion probabilities (sec. 6); *oclMPS+addU* - *oclMPS* with  $\Gamma\Delta Y$ ; *fullMPS* - *oclMPS* with  $\Gamma\Delta Y$  and the inter-people exclusion term  $\Omega$  (the full multi person model).

body parts are occluded (sec. 5); (iii) the pose estimation accuracy of our joint MPS model, compared to a state-of-the-art 1PS estimator [1] (sec. 6).

We split the group photo dataset into training (first 350 images) and testing (remaining 175 images). We use the training set to learn all the parameters. The test set is used for evaluating both the occlusion predictor and pose estimation performance.

**Automatic parameter setting.** In order to incorporate the occlusion probabilities in the MPS model (sec. 5.3) we need just two parameters (the scaling for the unary energy of the occlusion state and the real-to-occlusion state transition energy in the pairwise terms). We search over a grid of values and retain those maximizing the performance of the HPE algorithm (i.e. the Percentage of Correct body Parts) on the training set (sec. 9(iii)). The optimal weights between the various terms of MPS (eq. (8)) are learned using a constraint generation algorithm inspired by [23], again to maximize PCP. In the complete model, we train both types of parameters jointly (i.e. by running the constrain generation algorithm at every point on the grid). All other parameters of our model are learned as described in the respective sections 5.1, 5.2, 7, 8.

**Person Detector.** Since our approach relies on a person detector, we need one yielding high detection rates and low false positive rates, also on images where people are only visible from the waist up (Fig. 8). For this we combine a face detector [24] with the upper and full human body models in the detection framework of [25]. This detector achieves 86% detection-rate at 0.5 false positives per image on our group photo dataset.

**(i) FtB Orders Reduction.** Without any of the FtB orders reductions proposed in section 4, the median number of required pose estimations per image over the entire dataset is 600. When utilizing the exact reductions (sec. 4.1) this decreases to 80, and with also approximate reductions to 48 (sec. 4.2).



**Fig. 8. Results.** First column: top - results of single person model  $IPS+addU$ , bottom - full multi-person model  $fullMPS$ . Other columns: more results returned by our full model.

**(ii) Occlusion Prediction (OP).** Given a test image, we compute occlusion probabilities using eq. (9). This estimates a probability of occlusion  $\mathcal{O}_i^p$  for each person  $p$  and body part  $i$  in the image. We evaluate the quality of this estimation by using it to classify body parts as occluded or not-occluded. For this, we draw ROC curves by thresholding the probability at increasing values in  $[0, 1]$  (Fig. 7a).

Fig. 7a shows the performance of our method in 3 modes, differing in the amount of information given to the algorithm, and a few intuitive baselines to compare against. Our OP predictor in all modes clearly outperforms even the strongest baseline (*occlusion prior*). The influence of the order marginalization (eq. 9) on the prediction quality is visible by comparing *people+order* to *people*. This approximation only causes a modest performance drop. The influence of using our automatic (and imperfect) person detector can be seen by comparing *people* to *automatic*. The performance of the occlusion predictor decreases only marginally compared to using ground-truth detections. Finally, comparing *automatic* and *automatic-approxZreductOFF* demonstrates that the heuristics for reducing the number of FtB orders (sec. 4.2) do help the OP predictor. The good performance of the *automatic* mode shows that our predictor can reliably estimate the occlusion probabilities of persons' body parts given just their (automatically detected) image locations.

**(iii) Pose Estimation.** We evaluate the impact of our joint MPS model, which explicitly models interactions between people, on pose estimation performance. For each body part of every person, our method returns a line segment or deems the part as occluded. We evaluate performance using the framework of [1] (on-line) modified to account for occlusions. The performance is measured by average PCP (Percentage of Correctly estimated body Parts) over all persons correctly localized by the person detector. An estimated part is considered correct if its segment endpoints lie within a fraction of the length (*pcp-threshold*) of the ground-truth segment from their annotated location. An occluded body part is considered correct only if it is also occluded in the ground-truth. Fig. 7b shows PCP performance for *pcp-threshold* in  $[0.1, 0.5]$ .

We compare to, and based our model on, the single-person HPE of [1] with added the excellent body part models of [3] (*IPS*). This achieves sharper part posterior marginals than [1] alone, which is beneficial to our sampling procedure (sec. 6). We also compare

to the complete HPE of [3] using the code released by the authors<sup>1</sup>, initialized from the same person detections as 1PS and MPS. As Fig. 7b shows, extending *1PS* into our MPS by incorporating the occlusion probability prediction brings a substantial gain of 10% PCP (*1PS* vs *oclMPS*). Further adding the additional unary cues improves performance by another 2% (*oclMPS+addU*). Adding also the inter-people exclusion term  $\Omega$  brings another 2% improvement (*fullMPS*). This shows that all components presented in this paper are valuable for good performance in group photos. Overall, our full multi-person model improves over *1PS* by 15% (at  $\text{pcp-threshold} = 0.2$ ). Note how already our *1PS* outperforms [3] on this dataset. Fig. 8 shows some qualitative results (illustrations for the entire test set are available at [4]).

## 10 Conclusions

We presented a novel multi-person pose estimation framework that explicitly models interactions between people and estimates their poses jointly. Both occlusion probability and pose estimation evaluations confirm our claims that joint multi-person pose estimation in the group photo scenario is beneficial over estimating the pose of every person independently.

## References

1. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: BMVC (2009)
2. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Pose search: retrieving people using their pose. In: CVPR (2009)
3. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR (2009)
4. <http://www.vision.ee.ethz.ch/~calvin>
5. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61 (2005)
6. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS (2006)
7. Sigal, L., Black, M.: Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In: CVPR, vol. 2, pp. 2041–2048 (2006)
8. Lan, X., Huttenlocher, D.P.: A unified spatio-temporal articulated model for tracking. In: CVPR, vol. 1, pp. 722–729 (2004)
9. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. PAMI 28, 44–58 (2006)
10. Wang, Y., Mori, G.: Multiple tree models for occlusion and spatial constraints in human pose estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 710–724. Springer, Heidelberg (2008)
11. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. IJCV 75, 247–266 (2007)
12. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR (2008)
13. Ess, A., Leibe, B., Schindler, K., Gool, L.V.: Robust multi-person tracking from a mobile platform. PAMI 31(10), 1831–1846 (2009)

<sup>1</sup> We thank Andriluka and Schiele for help in evaluating their approach on our dataset.

14. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV (2009)
15. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML (2001)
16. Buehler, P., Everingham, M., Huttenlocher, D., Zisserman, A.: Long term arm and hand tracking for continuous sign language tv broadcasts. In: BMVC (2008)
17. Lan, X., Huttenlocher, D.: Beyond trees: Common-factor models for 2D human pose recovery. In: ICCV, vol. 1 (2005)
18. Gammeter, S., Ess, A., Jaeggli, T., Schindler, K., Van Gool, L.: Articulated multi-body tracking under egomotion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 816–830. Springer, Heidelberg (2008)
19. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI 28, 1568–1583 (2006)
20. Sutherland, I., Hodgman, G.: Re-entrant polygon clipping. Communications of the ACM (1974)
21. Buehler, P., Everingham, M., Huttenlocher, D.P., Zisserman, A.: Long term arm and hand tracking for continuous sign language TV broadcasts. In: BMVC (2008)
22. Jiang, H.: Human pose estimation using consistent max-covering. In: ICCV (2009)
23. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)
24. Froba, B., Ernst, A.: Face detection with the modified census transform. In: IEEE International Conference on Automatic Face and Gesture Recognition (2004)
25. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2009) (in press)

# Joint People, Event, and Location Recognition in Personal Photo Collections Using Cross-Domain Context\*

Dahua Lin<sup>1,2</sup>, Ashish Kapoor<sup>2</sup>, Gang Hua<sup>3</sup>, and Simon Baker<sup>2</sup>

<sup>1</sup> Computer Science and Artificial Intelligence Laboratory, MIT

<sup>2</sup> Microsoft Research

<sup>3</sup> Nokia Research Center Hollywood

**Abstract.** We present a framework for vision-assisted tagging of personal photo collections using context. Whereas previous efforts mainly focus on tagging people, we develop a unified approach to jointly tag across multiple domains (specifically people, events, and locations). The heart of our approach is a generic probabilistic model of context that couples the domains through a set of cross-domain relations. Each relation models how likely the instances in two domains are to co-occur. Based on this model, we derive an algorithm that simultaneously estimates the cross-domain relations and infers the unknown tags in a semi-supervised manner. We conducted experiments on two well-known datasets and obtained significant performance improvements in both people and location recognition. We also demonstrated the ability to infer event labels with missing timestamps (i.e. with no event features).

## 1 Introduction

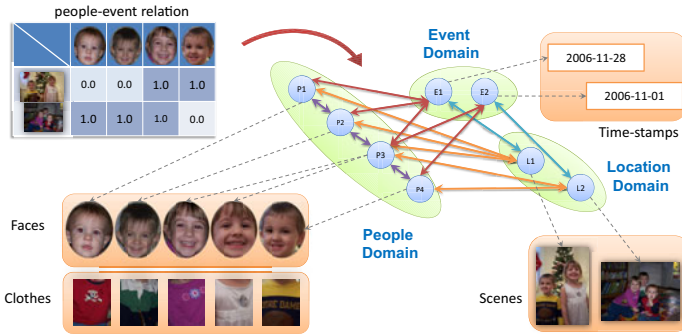
With the ever increasing popularity of digital photos, vision-assisted tagging of personal photo albums has become an active research topic. Existing efforts in this area have mostly been devoted to using face recognition to help tag people. However, current face recognition algorithms are still not very robust to the variation of face appearance in real photos. To address this issue, various methods [1] have been proposed to exploit contextual cues to aid recognition. While obtaining some improvement, these methods focus on the people domain, and neglect other important domains such as events and locations.

The most important questions in regard to personal photo tagging are *who*, *what*, *when*, and *where*. With an aim of answering these questions coherently, we consider the domains of people, events, and locations, as a whole. Our work is motivated by the insight that the domains are not independent and knowledge in one domain can help the others. For example, if we know the event that a photo was captured in, we can probably infer who was in the photo, or at least

---

\* The research described in this paper was conducted when all four authors were affiliated with Microsoft Research Redmond.





**Fig. 1.** Our framework comprises three types of entity: (1) The people, event, and location **domains**, together with their instances. (2) The **observed features** of each instance in each domain. (3) A set of contextual **relations** between the domains. Each relation is a 2D table of coefficients that indicate how likely a pair of labels is to co-occur. Although only the people-event relation is shown in this figure, we consider four different relations in this paper. See body of text for more details.

reduce the set of possibilities. On the other hand, the identities of the people in a photo may help us infer when and where the photo was taken.

Ideally, if a strong classifier is available to recognize the instances in a domain accurately, one can utilize the labels in this domain to help the recognition in others. However, a challenge arises in real system is that we often do not have a strong classifier to start with in any domain. One of our primary contributions is to develop a unified framework that couples the recognition in these domains. We also derive a joint learning and inference algorithm that would allow us to achieve accurate recognition in all domains by exploiting the statistical dependency between them to reinforce individual classifiers.

Our framework, outlined in figure 1, consists of three domains: people, events, and locations. Each domain contains a set of instances. In order to account for the uncertainty due to missing data or ambiguous features, we consider the labels in all three domains as random variables to be inferred. Pairs of domains are connected to each other through a set of cross-domain relations that model the statistical dependency between them.

In this paper, we specifically consider four relations: (a) the *people-event relation* models who attended which events, (b) the *people-people relation* models which pairs of people tend to appear in the same photo, (c) the *event-location relation* models which event happened where, and (d) the *people-location relation* models who appeared where. These relations embody a wide range of contextual information, which is modeled uniformly under the same mathematical framework. It is important to note that each pair of related domains are symmetric with respect to the corresponding relation. This means, for example, that utilizing the people-event relation, event recognition can help people recognition, and people recognition can also help event recognition.

Based on this framework, we formulate a joint probabilistic model to integrate both feature similarity and contextual relations. However, we face a challenge

that specially arises in the application of personal photo tagging. Unlike other classification problem such as object recognition where one can learn the contextual models from training data, the relational models (e.g. people-event relation) estimated from one photo collection are generally not applicable to other collections. In fact, the set of people or events may well be completely different in two different photo collections. It is also infeasible to require a user to prepare training data for each of their albums. Instead, we develop an algorithm that simultaneously estimates the relations and infers the labels across all domains by solving a unified optimization problem in a semi-supervised way.

We tested our approach on two well-known datasets. For people labeling, the error rate is reduced from 27.8% to 3.2% on one data set, and from 26.3% to 14.6% on the other. We also obtained a huge improvement in location labeling (16.7% to 1.3%). Finally, we demonstrate the ability to estimate event labels for photos in the presence of missing timestamps (i.e. with missing event features.)

## 2 Related Work

Related prior work can be roughly split into two categories: context-aided face recognition, and object/scene classification using context. We now review this related work and clarify the key differences from our approach.

Over the last decade, there has been a great deal of interest in the use of context to help improve face recognition accuracy in personal photos. A recent survey of context-aided face recognition can be found in [1]. Zhang et al. [2] utilized body and clothing in addition to face for people recognition. Davis et al. [3,4] developed a context-aware face recognition system that exploits GPS-tags, time-stamps, and other meta-data. Song and Leung [5] proposed an adaptive scheme to combine face and clothing features based on the time-stamps. These methods treat various forms of contextual cues as linearly additive features, and thus oversimplifies the interaction between different domains.

Various methods based on co-occurrence have also been proposed. Naaman et al. [6] leveraged time-stamps and GPS-tags to reduce the candidate list based on people co-occurrence and temporal/spatial re-occurrence. Gallagher and Chen [7] proposed an MRF to encode both face similarity and exclusivity. In later work by the same authors [8], a group prior is added to capture the tendency that certain groups of people are more likely to appear in the same photo. In addition, Anguelov et al. [9] developed an MRF model to integrate face similarity, clothing similarity and exclusivity. Finally, Kapoor et al. [10] proposed a framework that uses Gaussian Processes to capture contextual constraints. Whereas these models provide a more flexible way to capture the interaction between co-occurring instances, they are nearly all formulated within the people domain. An exception is Naaman et al. [6], which uses time and locations, however the model is heuristic and the time and location labels are treated as noiseless.

In contrast to prior contextual face recognition work, our framework treats all three domains in a uniform manner. The labels in each domain (including events and locations) are modeled as random variables, rather than noiseless quantities, and the relation connecting each pair of domains can be utilized for the

inference in both domains. Moreover, instead of using heuristics to utilize time and locations, we develop a principled approach that establishes a joint probabilistic model over these domains. Labeling and estimation are thus performed as a unified optimization process.

Our framework is also related to the use of context in object recognition and scene classification. For example, Torralba et al. [11,12] used scene context as a prior for object detection and recognition. Rabinovich et al. [13] proposed a CRF model that utilizes object co-occurrence to help object categorization. Galleguillos et al. [14] extended this framework to use both object co-occurrence and spatial configurations for image segmentation and annotation. Li-Jia and Fei-Fei [15] proposed a generative model that can be used to label scenes and objects by exploiting their statistical dependency. In later work [16], the same authors extended this model to incorporate object segmentation. Cao et al. [17] employed a CRF model to label events and scenes coherently.

While these approaches share some technical similarity with our work, three key differences distinguish our work:

(1) As mentioned above, it is infeasible in personal photo tagging to provide a separate training set to estimate the contextual model. To meet this challenge, we designed an algorithm where the model is estimated directly from the photo collection to be tagged, along with inference being performed. This should be contrasted with the conventional approach to object/scene classification, where the models are learned offline on a training set.

(2) The instances to be labeled in object/scene recognition are typically instances (e.g. objects) within a *single image*. The context models the relations (spatial, co-occurrence) within that image. On the other hand, our contextual model is over the *entire photo collection*. It models inter-photo dependencies rather than just intra-image relations. This makes it possible to reliably estimate the relational models without the need of a priori training.

(3) The application domain is different. Rather than considering generic object recognition and scene classification, we consider the problem of context-assisted face, location, and event recognition in personal photo collections.

### 3 Probabilistic Model Formulation

In this section, we formalize our framework as a Bayesian model. Suppose there are  $M$  domains:  $\mathcal{Y}_1, \dots, \mathcal{Y}_M$ . Each domain is modeled as a set of instances, where the  $i$ -th instance in  $\mathcal{Y}_u$  is associated with a label of interest, modeled as a random variable  $y_u^i$ . While the user can provide a small number of labels in advance, most labels are unknown and to be inferred. Specifically, we consider three domains for people, events, and locations. Each detected face corresponds to a person instance in people domain, and each photo corresponds to both an event instance and a location instance. Each domain is associated with a set of features to describe its instances. In particular, person instances are characterized by

their facial appearance and clothing; while events and locations are respectively characterized by time-stamps and the background color distribution.

To exploit the statistical dependency between the labels in different domains, we introduce a relational model  $R_{uv}$  between each pair of related domains  $\mathcal{Y}_u$  and  $\mathcal{Y}_v$ . It is parameterized by a 2D table of coefficients that indicate how likely a pair of labels is to co-occur. Taking advantage of these relations, we can use the information in one domain to help infer the labels in others.

Formally, our goal is to jointly estimate the posterior probability of the labels  $Y$  and relations  $R$  conditioned on the feature measurements  $\mathbf{X}$ :

$$p(Y, R|\mathbf{X}) \propto p(Y|R, \mathbf{X})p(R). \quad (1)$$

Here, we use  $Y$  and  $\mathbf{X}$  to represent the labels and features of all domains. The formulation has two parts: (1)  $p(Y|R, \mathbf{X})$ : the joint likelihood of the labels given the relational models and features (section 3.1). (2)  $p(R)$ : the prior put on the relations to regularize their estimation (section 3.2).

### 3.1 Joint Probability of Labels

We propose to directly model the joint label distribution conditioned on the observed features, rather than assuming a parametric feature distribution for each class as in generative models. This approach is generally more effective when the number of labeled samples in each class is limited. In particular, we propose the following model for  $p(Y|R, \mathbf{X})$ :

$$p(Y|\mathbf{X}; R) = \frac{1}{Z} \exp \left( \sum_{u=1}^M \alpha_u \Phi_u(Y_u; \mathbf{X}_u) + \sum_{(u,v) \in \mathcal{R}} \alpha_{uv} \Phi_{uv}(Y_u, Y_v; R_{uv}) \right). \quad (2)$$

The proposed likelihood contains: (1) an *affinity potential*  $\Phi_u(Y_u, \mathbf{X}_u)$  for each domain  $\mathcal{Y}_u$  to model feature similarity, and (2) a *relation potential*  $\Phi_{uv}(Y_u, Y_v; R_{uv})$  for each pair of related domains  $(u, v) \in \mathcal{R}$ . They are combined with weights  $\alpha_u$  and  $\alpha_{uv}$ , which can be set by cross-validation in practice.

**1. The affinity potential  $\Phi_u$**  captures the intuition that two instances in  $\mathcal{Y}_u$  with similar features are likely to be in the same class:

$$\Phi_u(Y_u; \mathbf{X}_u) = \sum_{i=1}^{N_u} \sum_{j=1}^{N_u} w_u(i, j) \mathbb{I}(y_u^i = y_u^j). \quad (3)$$

Here,  $w_u(i, j)$  is the similarity between the features of the instances corresponding to  $y_u^i$  and  $y_u^j$ .  $\mathbb{I}(\cdot)$  denotes the indicator that equals 1 when the condition inside the parenthesis holds. The similarity function  $w_u$  depends on the features used for that domain (see section 5 for details). If the instances in a domain can be described by different types of features, we define affinity potentials for different features, and use their sum as the overall potential.

Intuitively,  $\Phi_u$  considers all instances of  $\mathcal{Y}_u$  over the entire collection, and attains large value when instances with similar features are assigned the same

labels. Maximizing  $\Phi_u$  should therefore result in clusters of instances that are consistent with the the feature affinity. This is in contrast to standard CRF models [18] that require learning class-specific feature coefficients for each class.

When clothing is used as one of the features in the people domain, a modification is necessary. As people may change clothes, comparing clothing features is only appropriate when the two person instances were in the same event. To model this, we modify the affinity potential for clothing features to be:

$$\Phi(Y_P; \mathbf{X}_C) = \sum_{i=1}^N \sum_{j=1}^N w_C(i, j) \mathbb{I}(y_p^i = y_p^j) \mathbb{I}(y_e^{ph(i)} = y_e^{ph(j)}). \quad (4)$$

Here,  $Y_P$  and  $\mathbf{X}_C$  denote the people labels and clothing features,  $w_C(i, j)$  is the similarity between the clothes of the  $i$ -th and  $j$ -th person instances, and  $y_e^{ph(i)}$  and  $y_e^{ph(j)}$  are the event labels of the corresponding photos. The factor  $\mathbb{I}(y_e^{ph(i)} = y_e^{ph(j)})$  only turns on rest of the term within the same event.

**2.** The **relational potential**  $\Phi_{uv}(Y_u, Y_v; R_{uv})$  models the cross-domain interaction between the domains  $\mathcal{Y}_u$  and  $\mathcal{Y}_v$ . The relational model  $R_{uv}$  is parameterized as a 2D table of co-occurring coefficients between pairs of labels. For example, for people domain  $\mathcal{Y}_u$  and event domain  $\mathcal{Y}_v$   $R_{uv}(k, l)$  indicates how likely it is that the person  $k$  attended the event  $l$ . Then, we define  $\Phi_{uv}$  to be:

$$\Phi_{uv}(Y_u, Y_v; R_{uv}) = \sum_{i \sim j} \sum_{k, l} R_{uv}(k, l) \mathbb{I}(y_u^i = k) \mathbb{I}(y_v^j = l). \quad (5)$$

Here,  $i \sim j$  means that  $y_u^i$  and  $y_v^j$  co-occur in the same photo. Intuitively, large value of  $R_{uv}(k, l)$  indicate that the pair of labels  $k$  and  $l$  co-occur often, and will encourage  $y_u^i$  to be assigned  $k$  and  $y_v^j$  be assigned  $l$ . Hence, maximizing  $\Phi_u$  should lead to the labels that are consistent with the relation.

### 3.2 Relational Model Prior

In real application, only a relatively small number of instances are tagged in advance by user (often just one or two per class). The model is estimated from these user-given labels. While the estimation can also use the labels inferred in previous step in our iterative algorithm, the inferred labels could be noisy and actually depend on the user-given labels. To avoid over-fitting, it is important to regularize the relational models. To this end, we incorporate the following prior:

$$p(R) = \frac{1}{Z_{prior}} \exp \left( -\beta_1 \sum_{(u,v) \in \mathcal{R}} \|R_{uv}\|_1 - \beta_2 \sum_{(u,v) \in \mathcal{R}} \|R_{uv}\|_2^2 \right). \quad (6)$$

Here,  $\|R_{uv}\|_1$  and  $\|R_{uv}\|_2$  are L1 and L2 norm of the relational matrix. Intuitively, the first term encourages sparsity of the relational coefficients, and therefore can effectively suppress the coefficients due to occasional co-occurrences, retaining only those capturing truly stable relations. Furthermore, it is often the

case that a small number of people may appear hundreds of times, while others only several times. This could result in exceptionally large coefficients for those dominant classes, and as a consequence, some instances in small classes may be incorrectly assigned the labels of large classes. The second term regularizes the coefficients, and thus can help to inhibit such errors that could otherwise occur when class sizes are imbalanced.

## 4 Joint Inference and Learning

We derive a variational EM algorithm where the goal is to jointly infer the labels of instances and estimate the relational model. With a few labels in different domains provided in advance by user (denoted as  $Y_L$ ), the algorithm iterates between two steps: (1) Infer the distribution of the unknown labels (denoted as  $Y_U$ ) based on both the extracted features and the current relational model  $R$ . (2) Estimate and update the relational model  $R$  using the labels provided by user and the hidden labels inferred in previous iteration.

We can derive such iterative procedure by considering the task of Maximum-a-posteriori (MAP) estimation of  $R$

$$R^* = \operatorname{argmax}_R p(R|Y_L; \mathbf{X}), \quad \text{where } p(R|Y_L; \mathbf{X}) \propto p(R) \sum_{Y_U} p(Y_U, Y_L|R, \mathbf{X}). \quad (7)$$

Note that computing  $p(R|Y_L; \mathbf{X})$  requires marginalizing over the unknown labels  $Y_U$  and is intractable. The variational methods tackle this problem by maximizing a tractable lower a bound of the log posterior. Formally, if  $q$  denotes any valid distribution of  $Y_U$ , then using Jensen’s equality it is easy to obtain a lower bound of  $\log[p(R)p(Y_L|R, \mathbf{X})]$ , given by

$$J(R, q) = \mathbb{E}_q\{\log p(Y_U, Y_L|R, \mathbf{X})\} + \log p(R) + H_q(q(Y_U)) \quad (8)$$

Further, it is well known (put some ref here) that equality holds when  $q(Y_U) = p(Y_U|Y_L; R, \mathbf{X})$ . In other words, maximizing the lower bound  $J(R, q)$  with respect to both  $R$  and  $q$  will not only provide us with an estimate of  $R$  but also the posterior distribution over  $Y_U$ . The optimization of  $J(R, q)$  w.r.t.  $R$  and  $q$  can be performed by iterating between the following steps.

$$\hat{q}^{(t+1)} = \operatorname{argmax}_q J(\hat{R}^{(t)}, q), \quad \text{(E-step)} \quad (9)$$

$$\hat{R}^{(t+1)} = \operatorname{argmax}_R J(R, \hat{q}^{(t+1)}). \quad \text{(M-step)} \quad (10)$$

The E-step in Eq. (9) infers the posterior distribution of the unknown labels  $Y_U$  using the current model  $\hat{R}^{(t)}$ . The M-step in Eq. (10) estimates the relational model  $R$  based on the updated distribution  $\hat{q}^{(t+1)}(Y_U)$ . However, solving Eq. (9) and Eq. (10) under our formulation is intractable and we need to resort to variational approximations.

**Inferring Unknown Labels (E-STEP):** The optimization problem in Eq. (9) can be made tractable using *mean field approximation* [19]. Formally, we restrict

$q$  to be a factorized distribution:  $q(Y_U) = \prod_{u=1}^M \prod_{i \in U_u} q_u^i(y_u^i)$ . Here,  $U_u$  correspond to all unlabeled instances in domain  $\mathcal{Y}_u$ . The approximation results in the following closed form expressions for updating the posteriors:

$$\hat{q}_u^i(k) = \frac{1}{Z_u^i} \exp(\psi_u^i(k)). \quad (11)$$

where,  $Z_u^i = \sum_{k'} \exp(\psi_u^i(k'))$  is the normalization constant, and  $\psi_u^i(k)$  is:

$$\psi_u^i(k) = \alpha_u \sum_{j=1}^{N_u} w_u(i, j) q_u^j(k) + \sum_{v:(u,v) \in \mathcal{R}} \alpha_{uv} \sum_{j:i \sim j} \sum_{l=1}^{K_v} R_{uv}(k, l) q_v^j(l). \quad (12)$$

Note that despite the factorized form, the parameters of  $q_u^i$  for different instances are coupled to each other and effect each other. Further, as observed in Eq.(12), both feature similarity (first term) and cross-domain relations (second term) are utilized in the inference, leading to an estimate of the posterior that considers both within-domain and cross-domain information.

**Estimating Relational Model (M-STEP):** Given the inferred distribution  $q$ , we can estimate the relational model  $R$  by solving Eq.(10):

$$R^* = \operatorname{argmax}_R E_q \{ \log p(Y_L, Y_U | \mathbf{X}; R) \} - \log Z(\mathbf{X}; R) + \log p(R). \quad (13)$$

Note that the log-partition function  $\log Z(\mathbf{X}; R)$  is intractable here. We use *tree-reweighted approximation* [20] to make it tractable. The basic idea is to divide the original model into tractable sub-models, and replace  $\log Z(\mathbf{X}; R)$  with a convex combination of the log-partition functions of the sub-models. The substitution results in an upper bound of  $\log Z(\mathbf{X}; R)$  [20]. In particular, we divide the joint model into affinity models and cross-domain relations, leading to the following upper bound:

$$\sum_{u=1}^M \theta_u A_u + \sum_{u \leftrightarrow v} \theta_{uv} B_{uv}(R_{uv}/\theta_{uv}) \quad (14)$$

Here  $A_u$  is the log-partition of the affinity model for  $\mathcal{Y}_u$  that is independent of  $R$ , and  $B_{uv}$  is the log-partition of the cross-domain relation. The coefficients  $\theta_u$  and  $\theta_{uv}$  are the weights of the convex combination of the models. Such an approximation simplifies the maximization step and now each relation can be estimated respectively by solving:

$$R_{uv}^* = \operatorname{argmax}_{R_{uv}} E_q \{ \Phi_{uv}(Y_u, Y_v; R_{uv}) \} - \theta_{uv} B_{uv}(R_{uv}/\theta_{uv}) + \log p(R_{uv}). \quad (15)$$

For simplicity, we set the weights to be  $\theta_{uv} = 1/\#\text{relations}$ . The objective is concave with a unique optimum and we use L-BFGS algorithm [21] to solve it.

## 5 Experiments

There are two publicly available datasets that are commonly used to evaluate research in personal photo tagging, which we call *E-Album* [22] and *G-Album* [23].

Since ground-truth labels are not provided, we estimate ground-truth by manually tagging each detected face. We also manually tag the event and location of each photo. We excluded the photos without any detected faces, and those whose ground-truth event and location labels could not be determined, leaving a subset of each album. In particular, *E-Album* contains 108 photos taken at 21 locations in 19 events, and 19 different people with 145 detected faces. *G-Album* contains 312 photos taken at 117 events, and 13 different people with 441 detected faces. The two albums give rise to different challenges. The sizes of the people classes in the E-Album are more unbalanced, while the G-Album has many more events, each containing only a small number of photos.

Feature extraction was performed as follows. For the people domain, we used the facial features proposed in [24]. A color histogram was used for the clothing. The location of the clothing relative to the face was determined using a simple geometric rule. For events we used the time-stamps as features. For locations, we used a color histogram of the background scene. For each feature, a distance measure is required. For the face features, we followed the algorithm in [24]. For clothes and location features, we used the Earth-mover’s distance [25]. For events, we defined the distance to be 0 if the time-stamps were on the same day, and 1 otherwise. Finally, we need to compute the affinity weights  $w_u(i, j)$ . We experimented with a number of alternatives, and found that the best approach is to connect each unlabeled instance to just the closest  $K$  labeled instances, and set  $w_u(i, j) = \exp(-d^2(\mathbf{x}_i, \mathbf{x}_j)/\sigma^2)$ . The value of  $w_u(i, j)$  for the other instances is set to zero. Here  $d(\mathbf{x}_i, \mathbf{x}_j)$  is the distance between the features  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . We determined the optimal values of  $K$  and  $\sigma$  by cross validation.

Our algorithm outputs an estimate of the posterior probability of each label for each instance. To compute an error metric for our algorithm, we sort the candidate labels in terms of their posterior probabilities. We then compute rank- $k$  error rates, the proportion of unlabeled instances whose top  $k$  candidate labels are all incorrect. To evaluate our algorithm, we generate a pre-labeled subset for each album by random sampling. For the people domain, we randomly chose 19 instances (13%) for the E-Album, and 49 instances (11%) for the G-Album. Here, we require that at least one instance is pre-labeled for each class. However, this requirement can be readily removed using active learning (see section 5.5), by which one can introduce new labels interactively.

## 5.1 People Labeling

We compare the performance of four different variants of our algorithm: (1) using only people affinity (no contextual information), (2) with the people-people relation, (3) with the people-event relation, and (4) with both relations.

The results of quantitative evaluation are shown in Figure 2. We note three observations: First, on both albums the people-people relation alone provides only a limited improvement (rank-1 errors reduced from 27.8% to 27.0% for the E-Album). Second, the people-event relation gives a much bigger improvement (rank-1 errors reduced from 27.8% to 11.9% for the E-Album). Third, the combination of the people-event relation and the people-people relation yields



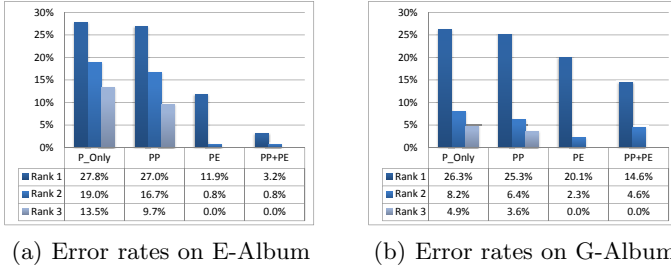


Fig. 2. Comparison of people labeling performance with different configurations.



Fig. 3. All rank-1 errors for the E-Album. Above the delimiter: Errors made by our algorithm with no contextual relations (27.8%). Below the delimiter: Errors made by our algorithm with both the people-event and people-people relations (3.2%).

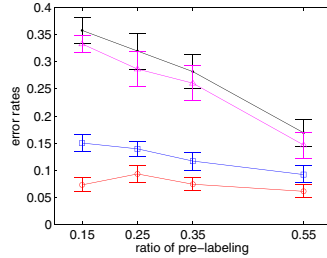
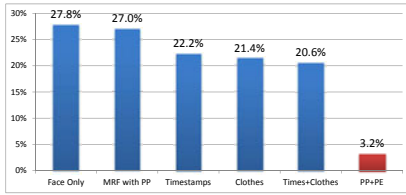


Fig. 4. The results of statistical significance testing obtained on E-Album with different percentages of pre-labeled instances. Curves from top to bottom obtained by: using only face, using people-people, using people-event, and using both relations.

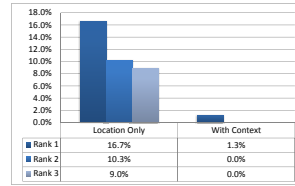
another significant improvement (rank-1 errors down to 3.2% on the E-Album). To illustrate our results visually, we include a collage of all of the errors for the E-Album in Figure 3. In the supplemental material, we include a similar figure for the G-Album, together with movies illustrating the results.

These results show: (1) that the people-event and people-people relations provide complementary sources of information, and (2) the people-event relation makes the people-people relation more effective than without it. The most likely explanation is that the group-prior and exclusivity are more powerful when used on the small candidate list provided by the people-event relation.

Overall, we found the G-Album to be more challenging. Partly, this is due to the fact that the G-Album contains a very large number of events (117), each with very few photos (3.8 on average.) The people-event relation would be more powerful with more photos per event. Note, however, that our framework still yields a substantial improvement, reducing the rank-1 error rate from 26.3% to 14.6%. Note also, that the rank-3 error rate is reduced to zero on both albums, a desirable property in vision-assisted tagging system where a short-list of candidates is often provided for the user to choose from.



**Fig. 5.** Comparison between baseline approaches and ours on E-Album



**Fig. 6.** Location error rates on E-Album

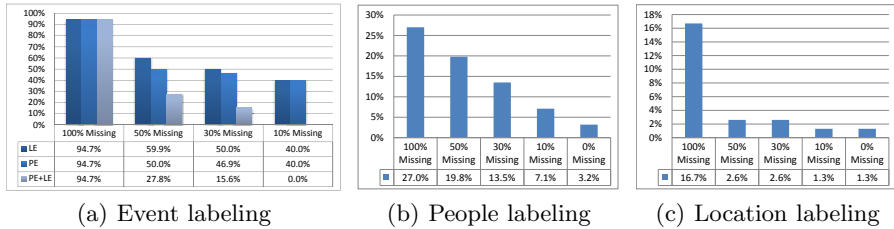
We also evaluated the performance of our framework with clothes features incorporated as conditional features. With both the people-event and people-people relations used, there are only three errors (3.2%) on the E-Album (see Figure 3). The clothing features are unable to correct any of these errors. For the G-Album, the conditional clothing features yield a slight improvement, with the best error rate reduced from 14.6% to 14.3%. The people-event and people-people relations are such powerful contextual cues that clothing adds little.

To validate the statistical significance of our results, we randomly generated multiple pre-labeled sets, with the percentage of pre-labeled instances varying from 15% to 55%. Figure 4 contains the median rank-1 results (signified by the central mark) along with the 25th and 75th percentiles (signified by lower and upper bars) obtained on E-Album. We also performed such testing on G-Album, and the results are provided in supplemental materials. The improvement is significant across the entire range of pre-labeling percentage in both data sets.

## 5.2 Comparison with Other Approaches

Direct comparison with published methods is difficult due to: (1) lack of a standard testing protocol, e.g. which instances are tagged in advance, and (2) different features were used in different papers, and the features used in prior work are not available. Hence, the most appropriate way to make a fair comparison with other approaches is to implement them and evaluate them using exactly the same data and features that we used. In particular, we compared with a combination of face feature and time-stamp cues (as in 3), a combination of face feature and clothes feature, and an adaptive combination of face feature and clothes feature conditioned on time stamps (as in 5). We also note that previous work that used an MRF to capture exclusivity and the group prior (e.g. 8) is essentially the special case of our framework where only the people-people relation is used. In all cases, we performed cross-validation to ensure that the best possible parameters were set for each particular algorithm.

Figure 5 contains the results on the E-Album. All of the feature-based algorithms yield a reasonable improvement with the rank-1 error rate being reduced from 27.8% to around 20% – 22%. While the MRF model using just the people-people relation (group prior and exclusivity) does not yield a notable reduction of rank-1 errors, it improves the rank-2 and rank-3 performance far more (the error



**Fig. 7.** The error rates of event, people and location labeling with varying percentages of missing time-stamps. For (b) the results were obtained with the P-E relation, and for (c) with the L-E relation.

rates are reduced from 19.0% and 13.5% to 16.7% and 9.7% respectively.) However, the performance improvement obtained by all of these methods is dwarfed by the improvement obtained by our algorithm when both the people-event and people-people relations are used (the rank-1 error is reduced to 3.2%).

Among the reasons that lead to such an improvement, the effective utilization of cross-domain context is the most important. Consider the people-event relation. When the event of a photo is inferred, the people classes that are not related to this event will be effectively ruled out from label selection (see Equations (5) and (12)), leaving only a very small subset of candidate labels to choose from. This resolves a great deal of ambiguity and makes recognition far easier.

### 5.3 Location Labeling

Figure 6 shows results for location estimation on the E-Album. We compare the results without any contextual information (location only) with those obtained using the event-location relation. The rank-1 error rate is reduced from 16.7% to 1.3%, and the rank-2 and rank-3 rates to 0%. Note that the event-location relation plays a similar role to the temporal priors used in video clustering [26].

### 5.4 Event Labeling with Missing Time-Stamps

The feature used for event labeling is the time-stamp of the photo. When present, this feature is very powerful; a temporal clustering of most photo collections breaks it naturally into events. In some cases time-stamps may be missing. For example, social networking sites such as Facebook remove timestamps. Furthermore, when merging two sets of photos collected on different cameras, it may not be wise to trust the time-stamps. In this section, we investigate what happens when time-stamps are missing.

We first investigated if we could estimate the event of a photo without the time-stamp. We randomly discarded 100%, 50%, 30%, and 10% of the time-stamps. The performance of event labeling under such conditions is shown in Figure 7(a). Note that we only compute the error rates over the photos without time-stamps. If all time-stamps are missing, we can only infer the event labels by random guessing, resulting in nearly 95% errors. If we know some of the

time-stamps, both event-location and people-event relations can be used to estimate a significant fraction of the event labels correctly. These two relations provide very complementary sources of information. The combination of the two is far better than either in isolation.

Next we investigated how the presence of missing time-stamps affects the performance of people and location labeling. In Figure 7(b) we see that the degradation in people-labeling performance with more and more missing time-stamps is very graceful. For location labeling, the removal of up to 50% of the timestamps hardly affects the performance. See Figure 7(c). So long as some photos captured in the same event retain their timestamps, the contextual benefit of the event-location relation is retained.

## 5.5 Labeling with Active Learning

As our framework estimates the posterior probabilities of the labels, it can be used for active learning [10]. By carefully choosing the order in which instances are pre-labeled, we can reduce the number of instances that need to be labeled to obtain a given recognition rate. We conducted preliminary experiments to illustrate this ability. In each iteration, we determine the unlabeled person instance that would lead to the maximum information gain and add it to the pre-labeled set. On average on the E-Album, it takes 30 iterations to obtain a rank-1 recognition rate of 95% for the people domain. In comparison, it requires 46 iterations with random sampling of the instances to be pre-labeled.

## 5.6 Timing Results

Our C# implementation runs in less than 2 seconds for both albums on a 2.0GHz Core-Duo laptop.

## 6 Conclusion

We have proposed the use of cross-domain relations as a mechanism to model context in multi-domain labeling (people, events, locations). Relation estimation and label inference are unified in a optimization algorithm. Our experimental results show that cross-domain relations provide a elegant, powerful, and general method of modeling context in vision-assisted tagging applications.

## References

1. Gallagher, A.C., Tsuhan, C.: Using context to recognize people in consumer images. *IPSN Journal* 49, 1234–1245 (2008)
2. Zhang, L., Chen, L., Li, M., Zhang, H.: Automated annotation of human faces in family albums. In: *11th ACM Conf. on Multimedia* (2003)
3. Davis, M., Smith, M., Canny, J., Good, N., King, S., Janakiraman, R.: Towards context-aware face recognition. In: *13th ACM Conf. on Multimedia* (2005)
4. Davis, M., Smith, M., Stentiford, F., Bamidele, A., Canny, J., Good, N., King, S., Janakiraman, R.: Using context and similarity for face and location identification. In: *SPIE'06* (2006)

5. Song, Y., Leung, T.: Context-aided human recognition - clustering. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 382–395. Springer, Heidelberg (2006)
6. Naaman, M., Garcia Molina, H., Paepcke, A., Yeh, R.B.: Leveraging context to resolve identity in photo albums. In: ACM/IEEE-CS Joint Conf. on Digi. Lib. (2005)
7. Gallagher, A.C., Tsuhan, C.: Using a markov network to recognize people in consumer images. In: ICIP (2007)
8. Gallagher, A.C., Chen, T.: Using group prior to identify people in consumer images. In: CVPR Workshop on SLAM'07 (2007)
9. Anguelov, D., Lee, K.c., Gokturk, S.B., Sumengen, B.: Contextual identity recognition in personal photo albums. In: CVPR'07 (2007)
10. Kapoor, A., Hua, G., Akbarzadeh, A., Baker, S.: Which faces to tag: Adding prior constraints into active learning. In: ICCV'09 (2009)
11. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: ICCV'03 (2003)
12. Torralba, A.: Contextual priming for object detection. *Int'l. J. on Computer Vision* 53, 169–191 (2003)
13. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV'07 (2007)
14. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: CVPR'08 (2008)
15. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: CVPR'07 (2007)
16. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding: Classification, annotation, and segmentation in an automatic framework. In: CVPR'09 (2009)
17. Cao, L., Luo, J., Kautz, H., Huang, T.S.: Annotating collections of photos using hierarchical event and scene models. In: CVPR'08 (2008)
18. Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields for Relational Learning. In: *Introduction to Statistical Learning*. MIT Press, Cambridge (2007)
19. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* 1, 1–305 (2008)
20. Wainwright, M.J., Jaakkola, T., Willsky, A.: A new class of upper bounds on the log partition function. *IEEE Transaction on Information Theory* 51, 2313–2335 (2005)
21. Byrd, R.H., Lu, P., Nocedal, J.: A limited memory algorithm for bound constrained optimization. *SIAM Journal on SSC* 16, 1190–1208 (1995)
22. Cui, J., Wen, F., Xiao, R., Tian, Y., Tang, X.: Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In: SIGCHI, pp. 367–376 (2007)
23. Gallagher, A.C.: Clothing cosegmentation for recognizing people. In: CVPR'08 (2008)
24. Hua, G., Akbarzadeh, A.: A robust elastic and partial matching metric for face recognition. In: ICCV'09 (2009)
25. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. *Int'l. Journal on Computer Vision* 40, 99–121 (2000)
26. Schroff, F., Zitnick, C., Baker, S.: Clustering videos by location. In: *British Machine Vision Conference* (2009)

# Chrono-Gait Image: A Novel Temporal Template for Gait Recognition

Chen Wang<sup>1</sup>, Junping Zhang<sup>1,\*</sup>, Jian Pu<sup>1</sup>,  
Xiaoru Yuan<sup>2</sup>, and Liang Wang<sup>3,4</sup>

<sup>1</sup> Shanghai Key Lab of Intelligent Information Processing  
School of Computer Science, Fudan University, China

<sup>2</sup> Key Laboratory of Machine Perception (Ministry of Education)  
School of Electronics Engineering and Computer Science  
Peking University, Beijing 100871, China

<sup>3</sup> Department of Computer Science, University of Bath, BA2 7AY, United Kingdom

<sup>4</sup> National Laboratory of Pattern Recognition, Institute of Automation,  
Chinese Academy of Sciences, China

chen.wang0517@gmail.com, jpzhang@fudan.edu.cn, mydaiyu@hotmail.com,  
xiaoru.yuan@pku.edu.cn, lw356@cs.bath.ac.uk

**Abstract.** In this paper, we propose a novel temporal template, called Chrono-Gait Image (CGI), to describe the spatio-temporal walking pattern for human identification by gait. The CGI temporal template encodes the temporal information among gait frames via color mapping to improve the recognition performance. Our method starts with the extraction of the contour in each gait image, followed by utilizing a color mapping function to encode each of gait contour images in the same gait sequence and compositing them to a single CGI. We also obtain the CGI-based real templates by generating CGI for each period of one gait sequence and utilize contour distortion to generate the CGI-based synthetic templates. In addition to independent recognition using either of individual templates, we combine the real and synthetic temporal templates for refining the performance of human recognition. Extensive experiments on the USF HumanID database indicate that compared with the recently published gait recognition approaches, our CGI-based approach attains better performance in gait recognition with considerable robustness to gait period detection.

## 1 Introduction

Biometric authentication is useful in many applications such as social security, individual identification in law enforcement and access control in surveillance. Compared with other biometric features such as face, iris and fingerprint, the advantages of gait include: 1) the acquisition of gait data is non-contactable, non-invasive, and hidden; 2) gait is the only perceptible at a distance. However, the performance of gait recognition suffers from some exterior factors such as

---

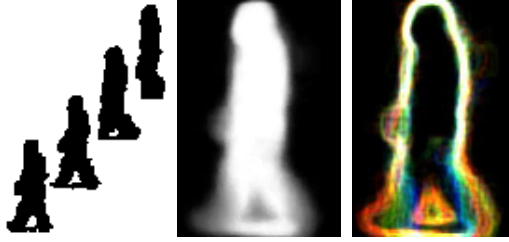
\* Corresponding author.

shoes, briefcases, clothing, and environmental context. Moreover, it depends on whether the spatio-temporal relationship between gait frames in a gait sequence can be effectively represented. Although it is a challenging task, the nature of gait indicates that it is an irreplaceable biometric [1] and can benefit remote biometric authentication.

## 1.1 Related Work

The extraction of gait features plays a crucial role in improving the performance of gait recognition. There are two main extraction methods: model-based and model-free approaches. Model-based approaches recover the underlying mathematical construction of gait with a structure/motion model [2]. Wang et al. adopted procrustes analysis to capture the mean shapes of the gait silhouettes [3]. However, procrustes analysis is time consuming and vulnerable to noise. Veres et al. [4] and Guo and Nixon [5] employed the analysis of variance and mutual information respectively to discuss the effectiveness of features for gait recognition. Bouchrika and Nixon proposed a motion-based model by using the elliptic Fourier descriptors to extract crucial features from human joints [6]. Wang et al. [7] combined structural-based and motion-based models by employing a condensation framework to refine the feature extraction. Although the structure-based models can to some degree deal with occlusion and self-occlusion as well as rotation, the performance of the approaches suffers from the localization of the torso and it is not easy to extract the underlying model from gait sequences [2,6]. Furthermore, it is necessary to understand the constraints of gait such as the dependency of neighboring joints and the limit of motion to develop an effective motion-based model [2].

As for the model-free approaches, we can divide them into two major categories based on the manners of preserving temporal information. The first strategy keeps temporal information in recognition (and training) stage [8,9,10,11]. Sundaresan et al. utilized a hidden Markov models (HMMs) based framework to achieve gait recognition [9]. Sarkar et al. [11] utilized the correlation of sequence pairs to preserve the spatio-temporal relationship between the gallery and probe sequences. Wang et al. [8] applied principal component analysis (PCA) to extract statistical spatio-temporal features of gait frames. However, large-scale training samples are generally needed for probabilistic temporal modelling methods (such as HMMs) to obtain a good performance. A disadvantage for the direct sequence matching methods is the high computational complexity of sequence matching during recognition and the high storage requirement of the dataset. The second strategy converts a sequence of images into a single template [11,12,13,14,15]. Liu et al. [12] proposed to represent the human gait by averaging all the silhouettes. Motivated by their work, Han and Bhanu [1] proposed the conception of gait energy image (GEI), and constructed the real and synthetic gait templates to improve the accuracy of gait recognition. Bashir et al. [16] also explored the invariant gait subspaces based on entropy. With a series of grayscale averaged gait images, Xu et al. employed discriminant analysis with tensor representation (DATER) for individual recognition [13]. Chen et al. proposed multilinear



**Fig. 1.** From left to right: a gait sequence, gait energy image, and chrono-gait image

tensor-based non-parametric dimension reduction (MTP) [15] for gait recognition. However, the above template-based methods lose the temporal information of gait sequences more or less. For example, averaging template methods throw out all the temporal order information of the gait sequence. Moreover, the time and space computational complexities of those tensor-based approaches are too high to be employed in real applications [13, 14, 15].

## 1.2 Our Contribution

In the recent years, the visualization community has studied how to effectively represent a sequence of images with a single colored image. For displaying time-varying data, especially for volumetric data, Woodring and Shen [17] investigated several different color-mapping strategies by encoding the time varying information of the data into color spectrum. Jänicke et al. [18] measured local statistical complexity for multifield visualization. More recently, Wang et al. [19] claimed that critically important areas are the most essential aspect of time-varying data to be detected and highlighted. However, it is difficult to directly employ such methods to generate a good temporal template for gait recognition since these algorithms are inefficient to compress gait sequences (in which gaits always have large overlapped regions between frames) into a 2-dimensional gait image.

Considering the pros and cons of gait recognition methods mentioned before, we focus on the single-template method in this paper because of its simplicity and low computational complexity. But in order to well preserve temporal information of gait patterns, we borrow some ideas from color temporal encoding in the visualisation community. In brief, we propose a novel temporal template to encode a gait sequence to a color image, named as Chrono-Gait Image (CGI). To further improve the discriminant ability of CGIs, we also propose a simple strategy to generate real and synthetic templates. An example of a gait sequence, GEI and CGI is shown in Fig. 1. In comparison with the state-of-the-art methods, our major contributions are: 1) Simple and easy to implement, CGI effectively preserves the temporal information in a gait sequence with a single-template image. 2) Unlike intensity, color, which has higher variance than grayscale, can enlarge the distance between gait sequences from different subjects and thus benefit gait recognition. 3) CGI is robust to different gait period detection methods which are



usually a basis of constructing gait templates. 4) To the best of our knowledge, color encoding gait images as a temporal template for gait recognition is not yet exploited in the biometric authentication community. Experiments indicate that compared with several recently published approaches, the CGI temporal template attains competitive performance on USF HumanID benchmark database.

The remainder of the paper is organized as follows. The proposed CGI temporal template is detailed in Section 2. We discuss the generation of real and synthetic CGI templates and the corresponding human recognition procedure in Section 3. Experiments are provided and analyzed in Section 4. Section 5 concludes the paper.

## 2 Chrono-Gait Images

In this paper, we attempt to achieve individual recognition under a particular human motion. Note that the motion, regular human walking, is generally used in most of current approaches of individual recognition by gait.

### 2.1 Motivation and Pre-processing

Because of the basic structure of human body, regular human walking always has a fixed cycle with a particular frequency. However, some methods lose the gait cycle information when performing a individual recognition by gait. Meanwhile, other methods need high computational cost for preserving such information. To address the issue, several fundamental assumptions in this paper are: 1) most of normal people have a similar gait gesture such as the stride length. 2) each person has his/her unique gait behavior, such as the shape of the torso, the moving range of limbs, and so on; 3) color can be used as a function of time. Under these assumptions, we can encode time-varying gait cycle information into a single chrono-gait image by color.

To obtain the CGIs, we directly employ the silhouette images that are extracted by the baseline algorithm proposed by Sarkar et al. [11]. Then we encode temporal information in the silhouette images with additional colors to generate a chrono-gait image. The goal of CGIs is to compress the silhouette images into a single image without losing too much temporal relationship between the images.

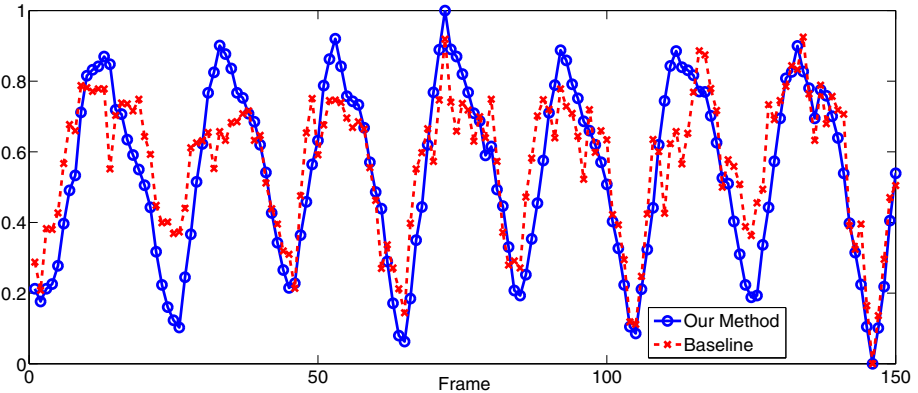
### 2.2 Period Detection

Regular human walking is a periodical motion. To preserve the temporal information, we need to detect the period in the gait sequence firstly. We propose to calculate average width  $W$  of the leg region in a gait silhouette image  $I$  as follows:

$$W = \frac{1}{\beta H - \alpha H} \sum_{i=\alpha H}^{\beta H} (R_i - L_i), 0 \leq \alpha \leq \beta \leq 1 \quad (1)$$

where  $H$  is the height of an image,  $L_i$  and  $R_i$  are the positions of the leftmost and rightmost foreground pixels in the  $i$ th line of the silhouette images, respectively.

To alleviate the influence of some exterior factors such as briefcase, shadow and surface that might be misclassified into the silhouette image, parameters  $\alpha$  and  $\beta$  are used to constrain the computation of the gait period to the leg region.

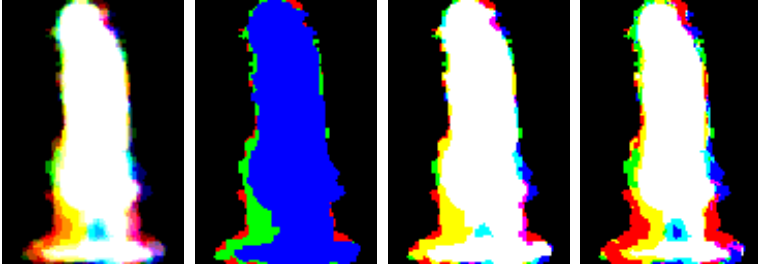


**Fig. 2.** Comparison between our method and baseline algorithm on gait period detection. The  $X$ -axis denotes the order of gait frames. The  $Y$ -axis represents the average width of each frame for our method, and the number of foreground pixels in the lower half of the silhouette for baseline method. Both of them are normalized to  $[0, 1]$ .

Sarkar et al. [11] proposed to detect such key frames by counting the number of foreground pixels in the lower half of the silhouettes in their baseline algorithm. A comparison of these two detection methods is illustrated as in Fig. 2, from which it can be seen that two detection methods pay attention to different parts of gait sequence. In the proposed period detection method, the average width  $W$  will have a local maximum when the two legs are farthest apart from each other and reach a local minimum when the two legs wholly overlap. Fig. 2 also indicates that our method produces sharper peaks and valleys, and thus preserves the correct temporal order well compared with the baseline algorithm [11].

### 2.3 Color Mapping

It is worth mentioning that in the visualization community, Woodring and Shen [17] used pseudo-color to visualize time-varying information for volume rendering. They proposed four integration functions: 1) Alpha Compositing; 2) First Temporal Hit; 3) Additive Colors; 4) Minimum/Maximum Intensity. However, their methods can not directly be applied to generate a temporal template since they assume that there is little overlapped foreground region between continuous frames, while in our case, the overlap of foreground silhouettes is serious between gait frames. Some results using their recommended four functions are illustrated in Fig. 3. All these images are generated from the same gait sequence. Due to the serious overlap between gait frames, we can see that the resulting



**Fig. 3.** Some results for visualizing gaits using Woodring and Shen’s algorithm [17]. a) Alpha Compositing; b) First Temporal Hit; c) Additive Colors; d) Minimum/Maximum Intensity.

images lose many important features. Naturally, such results are not truly informative to represent gait patterns.

Since the outer contour of the silhouette images is an important feature [3,8], and preserve the spatial information with small degree of overlap, we thus attempt to extract the contours instead of silhouettes. There are various edge detection techniques such as gradient operator, LoG operator and local information entropy [20], to extract the contours of the silhouette images. We use local information entropy to obtain the gait contour since it provides more abundant features than gradient and LoG operators. The local information entropy is defined as:

$$h_t(x, y) = -\left(\frac{n_0}{|\omega_d(x, y)|} \ln \frac{n_0}{|\omega_d(x, y)|} + \frac{n_1}{|\omega_d(x, y)|} \ln \frac{n_1}{|\omega_d(x, y)|}\right) \quad (2)$$

where the  $d$ -neighborhood of point  $(x, y)$  based on  $D_8$  distance (chessboard distance) is  $\omega_d(x, y) = \{(u, v) | \max\{|u - x|, |v - y|\} \leq d\}$ , and  $n_0$  and  $n_1$  are the numbers of foreground pixels and background pixels, respectively. Term  $t$  represents the frame label, and  $x$  and  $y$  denote the values in the 2-d image coordinate.

Then we normalize the entropy by the following formula:

$$h'_t(x, y) = \frac{h_t(x, y) - \min_{x,y} h_t(x, y)}{\max_{x,y} h_t(x, y) - \min_{x,y} h_t(x, y)}. \quad (3)$$

We also propose a liner interpolation function to encode the temporal information to three color components (R=Red, G=Green, B=Blue) as follows:

$$R(k_t) = \begin{cases} 0 & k_t \leq 1/2, \\ (2k_t - 1)I & k_t > 1/2 \end{cases} \quad (4)$$

$$G(k_t) = \begin{cases} 2k_t I & k_t \leq 1/2, \\ (2 - 2k_t)I & k_t > 1/2 \end{cases} \quad (5)$$

$$B(k_t) = \begin{cases} (1 - 2k_t)I & k_t \leq 1/2, \\ 0 & k_t > 1/2 \end{cases} \quad (6)$$

where  $k_t = (W_t - W_{min}) / (W_{max} - W_{min})$ .  $W_i$  represents the degree of two legs apart from each other, which is explained in Equ. 1.  $W_{max}$  and  $W_{min}$  are

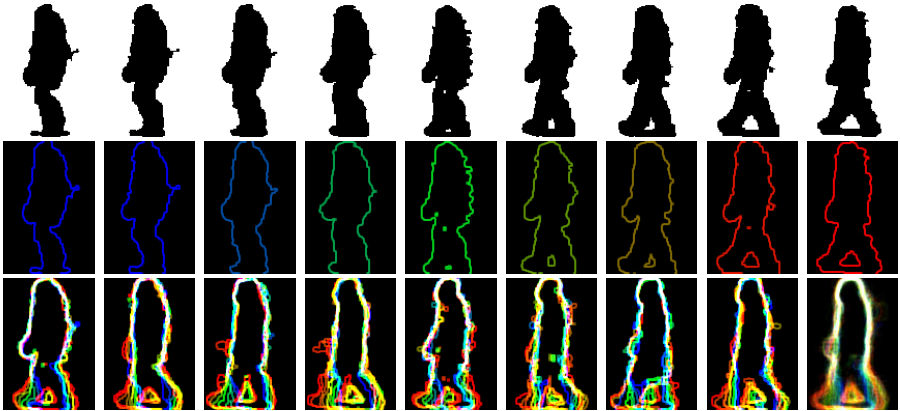


Fig. 4. An example of generating a CGI temporal template

the extreme widths of the period which the  $i$ th frame belongs to, and  $I$  is the maximum of intensity value, e.g., 255.

## 2.4 Representation Construction

We calculate the colored gait contour image  $\mathbf{C}_t$  of the  $t$ th frame in the gait sequence as:

$$\mathbf{C}_t(x, y) = \begin{pmatrix} h'_t(x, y) * R(k_t) \\ h'_t(x, y) * G(k_t) \\ h'_t(x, y) * B(k_t) \end{pmatrix} \quad (7)$$

Given the colored gait contour images  $\mathbf{C}_t$ , a CGI temporal template  $\mathbf{CG}(x, y)$  is defined as follows:

$$\mathbf{CG}(x, y) = \frac{1}{p} \sum_{i=1}^p \mathbf{PG}_i(x, y), \quad (8)$$

where  $p$  is the number of 1/4 gait periods, and  $\mathbf{PG}_i(x, y) = \sum_{t=1}^{n_i} \mathbf{C}_t(x, y)$  is the sum of the total  $n_i$  colored contour images in the  $i$ th 1/4 gait period.

The whole process to generate CGI is shown in Fig. 4. The first row shows 9 silhouettes in the  $\alpha$ th 1/4 gait period. And the second row shows the corresponding colored gait contour images after edge detection and color mapping. Then we sum all these 9 images to obtain the first one  $\mathbf{PG}_\alpha$  in the third row, representing this 1/4 period. The second to the eighth images on the third row represent  $\mathbf{PG}$ s corresponding to other different 1/4 periods in the same gait sequence. At last, we average all these frames to get the final CGI shown as the last one in the third row. It is not difficult to see that we obtain better visualization result and more informative gait template (which will be demonstrated in gait recognition experiments).

### 3 Human Recognition Using CGI Template

To employ the proposed CGI temporal template for individual recognition, one way is to directly measure the similarity between the gallery and probe templates. However, there are probably several disadvantages of doing so: 1) the templates obtained from the gait sequences may lead to overfit since such sequences are collected under similar physical conditions; 2) the number of CGIs is too small to characterize the topology of essential gait space; 3) when one pixel is viewed as one dimension, we will face to the problem of curse of dimensionality. Therefore, one solution to this is to generate two types of templates, namely real templates and synthetic templates. Meanwhile, we can project the templates into certain low-dimensional discrimination subspace with the dimension reduction method.

Specifically, we generate the real templates by referring to the colored image of each period (i.e., averaging continuous 4 PGs in one period) as a temporal template. One advantage is that such a template keeps the similar gait temporal information as the CGI of the whole sequence owns. Synthetic templates are used to enhance the robustness to the exterior factors such as the noise of shadow. Similar to Han and Bhanu [11], we cut the bottom  $2 \times i$  rows from the CGI and resize to the original size using the nearest neighbor interpolation. If parameter  $i$  varies from 0 to  $K - 1$ , then a total of  $K$  synthetic templates will be generated from each CGI template.

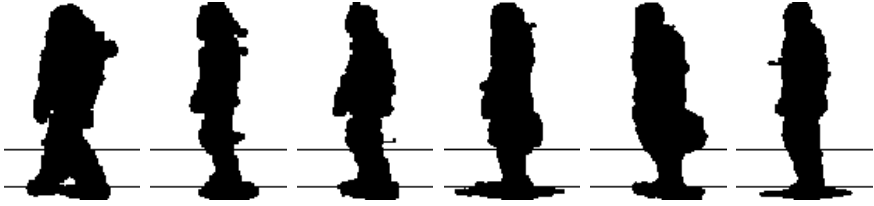
To address the curse of dimensionality issue, we employ Principal Component Analysis and Linear Discriminant Analysis (PCA+LDA) to project the real and synthetic templates in the gallery set into a low-dimensional subspace. And the real/synthetic templates in the probe set will be mapped into the low-dimensional subspace by using the projection matrix obtained by PCA+LDA. Let  $\hat{\mathcal{R}}_p$  and  $\hat{\mathcal{S}}_p$  be the real and synthetic templates of the individual in probe sets, respectively, and let  $\mathcal{R}_i$  and  $\mathcal{S}_i$  be the real and synthetic templates of the  $i$ th individual in the gallery sets, respectively. In the subspace, the real/synthetic templates are identified based on the minimal Euclidean distances ( $d(\hat{\mathcal{R}}_p, \mathcal{R}_j)$  or  $d(\hat{\mathcal{S}}_p, \mathcal{S}_j)$ ) between the probe real/synthetic feature vectors to the class center of the gallery real/synthetic feature vectors. To further improve the performance, we fuse the results of these two types of templates using the following equation:

$$d(\hat{\mathcal{R}}_p, \hat{\mathcal{S}}_p, \mathcal{R}_i, \mathcal{S}_i) = \frac{d(\hat{\mathcal{R}}_p, \mathcal{R}_i)}{\min_j d(\hat{\mathcal{R}}_p, \mathcal{R}_j)} + \frac{d(\hat{\mathcal{S}}_p, \mathcal{S}_i)}{\min_j d(\hat{\mathcal{S}}_p, \mathcal{S}_j)}, \quad i, j = 1, \dots, C \quad (9)$$

where  $C$  is the number of classes, i.e., the number of subjects here. We assign the probe template to the  $k$ th class if:

$$k = \arg \min_i d(\hat{\mathcal{R}}_p, \hat{\mathcal{S}}_p, \mathcal{R}_i, \mathcal{S}_i), \quad i = 1, \dots, C \quad (10)$$

More details about real and synthetic templates can be referred in Han and Bhanu's work [11]. Note that although we use distance measurements and fusion functions different from those used in [11], extensive experiments indicate that



**Fig. 5.** Some gait examples with two additional lines. The upper line is at  $3/4H$  and the lower one is at  $15/16H$  ( $H$  is the height of a gait image). All of these cases are collected from different probe sets. The fourth and fifth images are with briefcase and others are without briefcase.

the differences between the proposed fusion criterion and theirs are quite minor with respect to recognition accuracy.

## 4 Experiments

We evaluate the CGI algorithm on the USF HumanID gait database (silhouette version 2.1) [11]. The gait database consists of 122 individuals walking in elliptical paths on concrete and grass surface, with/without a briefcase, wearing different shoes, and sampling in elapsed time. Sarkar et al. [11] selected the sequences with “Grass, Shoe Type A, Right Camera, No Briefcase, and Time  $t_1$ ” for the gallery set, and developed 12 experiments, each of which is under some specific conditions. More details can be referred to Sarkar et al.’s work [11]. Because the USF database has provided the silhouette images after background subtraction and image alignment, all of our experiments are based on these silhouette images.

We evaluate the “Rank1” and “Rank5” performances of several recent approaches including baseline algorithm (based on silhouette shape matching) [11], GEI [1], HMM [21], IMED+LDA [14], 2DLDA [14], DATER [13], MTP [15] and Tensor Locality Preserving Projections (TLPP) [22]. The Rank1 performance means the percentage of the correct subjects ranked first while the Rank5 performance means the percentage of the correct subjects appeared in any of the first five places in the rank list. We also report the average performance by computing the ratio of correctly recognized subjects to the total number of subjects.

In Section 2.2, we introduce two parameters  $\alpha$  and  $\beta$ . For the USF HumanID database, we choose  $\alpha = 3/4$  and  $\beta = 15/16$ . Some examples in the database are shown in Fig. 5. From the figure, we can see that most of the briefcase is above the first  $3/4H$  line and most of the shadow is under the  $15/16H$  line. Therefore, it means that the influence of briefcase and shadow can be effectively decreased.

To evaluate the performance of the proposed CGI temporal template, we employ a simple 1-nearest neighbor classifier (1-NN) on the original GEI and CGI without using real/synthetic templates and PCA/LDA. We also provide the performance of baseline algorithm [11] for comparison. The neighborhood parameter  $d$  introduced in Section 2.3, which is used to describe the size of locality, is set to

**Table 1.** Comparison of Recognition Performance on USF HumanID Database using 1-NN. Here, V–View, S–Shoe, U–Surface, B–Briefcase, T–Time, C–Clothing

Exp.	Gallery Size	Difference	Rank1 Performance (%)			Rank5 Performance (%)		
			baseline [11]	GEI	CGI	baseline	GEI	CGI
A	122	V	73	84	<b>89</b>	88	93	<b>98</b>
B	54	S	78	87	<b>91</b>	93	<b>94</b>	<b>94</b>
C	54	SV	48	72	<b>74</b>	78	93	<b>94</b>
D	121	U	<b>32</b>	19	20	<b>66</b>	45	40
E	60	US	<b>22</b>	18	18	<b>55</b>	53	43
F	121	UV	<b>17</b>	10	11	<b>42</b>	29	23
G	60	USV	<b>17</b>	13	13	<b>38</b>	37	32
H	120	B	61	56	<b>76</b>	85	77	<b>90</b>
I	60	SB	57	55	<b>75</b>	78	77	<b>93</b>
J	120	VB	36	40	<b>57</b>	62	69	<b>80</b>
K	33	TSC	3	<b>9</b>	6	12	15	<b>24</b>
L	33	TUSC	3	3	<b>9</b>	15	15	<b>24</b>
Avg.			40.96	41.13	<b>48.33</b>	64.54	61.38	<b>65.45</b>

1 in our experiment. The results are summarized in Tab. 1. It can be seen from the Tab. 1 that 1) CGI achieves the best average performance among all the algorithms. 2) As illustrated in Exp. H, I, J, the performance of CGI is very robust to the briefcase condition, and in such conditions, the accuracy is improved by almost 20% compared with GEI. 3) Compared with GEI, CGI has better Rank5 performance than GEI in 8 out of 12 specific conditions. 4) In all the remaining 4 conditions, both GEI and CGI perform worse than the baseline algorithm in the surface condition. We can infer that the gait templates are more sensitive to the surface condition than the baseline algorithm because of the shadows or some other factors.

To discover which components of the proposed CGI temporal templates are crucial to the performance of gait recognition, we compare several variants of the contour-based temporal template with the silhouette-based template, which is employed by most of the gait recognition systems [13]. Here GEI-contour means that we compute the GEI based on contour images, and CGI-gray means that we average each CGI into a grayscale image.

We also employ the fusion of real and synthetic templates introduced in Section 3 to further improve the performance. To make the experiment fair, we use the same strategy to generate real and synthetic templates, assigning the same parameters to PCA and LDA to reduce the data set into a subspace. The fusion results are obtained using the same formula. More precisely, the reduced dimension is obtained when using 99% as the PCA cumulative contribution rate and the regularization parameter of LDA is set to  $10^8$  for real templates. And for synthetic templates, we set parameter  $K$  to 6, in other words, we cut the last 10 rows of each gait image to generate 6 synthetic templates. We use 99.5% as the PCA cumulative contribution rate and set the LDA parameter to 0. The size of locality  $d$  is also set to 1 here.

**Table 2.** Comparison of Recognition Performance of GEI, GEI-contour, CGI-gray, CGI using the Same Experiment Settings

Exp.	Rank1 Performance (%)				Rank5 Performance(%)			
	GEI	GEI-contour	CGI-gray	CGI	GEI	GEI-contour	CGI-gray	CGI
A	87	85	86	<b>92</b>	96	96	<b>97</b>	96
B	<b>93</b>	91	<b>93</b>	<b>93</b>	94	<b>96</b>	<b>96</b>	94
C	74	72	<b>81</b>	76	<b>94</b>	93	<b>94</b>	93
D	34	27	38	<b>47</b>	66	52	70	<b>75</b>
E	38	37	42	<b>48</b>	63	60	68	<b>70</b>
F	21	11	31	<b>34</b>	47	31	53	<b>54</b>
G	23	17	28	<b>40</b>	47	42	<b>58</b>	57
H	57	74	72	<b>82</b>	81	93	93	<b>94</b>
I	58	73	<b>75</b>	73	80	88	87	<b>93</b>
J	52	57	59	<b>62</b>	78	82	82	<b>83</b>
K	<b>9</b>	<b>9</b>	<b>9</b>	6	21	<b>30</b>	27	27
L	6	12	9	<b>15</b>	24	<b>36</b>	27	24
Avg.	49.06	49.90	55.74	<b>60.54</b>	70.46	69.52	75.89	<b>76.83</b>

**Table 3.** Comparison of Recognition Performance of GEI, CGI using two period detection method. We refer the period detection method proposed in Sarkar et al. [11] as “C” and our method proposed in section 2.2 as “W”.

Exp.	Rank 1 Performance (%)				Rank 5 Performance (%)			
	GEI+C	GEI+W	CGI+C	CGI+W	GEI+C	GEI+W	CGI+C	CGI+W
A	87	88	<b>85</b>	<b>92</b>	96	95	93	96
B	93	91	<b>87</b>	<b>93</b>	94	96	94	94
C	74	76	78	76	94	93	93	93
D	<b>34</b>	<b>40</b>	47	47	66	66	72	75
E	38	38	53	48	63	63	70	70
F	21	25	30	34	47	45	55	54
G	23	28	37	40	47	50	55	57
H	57	58	<b>76</b>	<b>82</b>	81	78	93	94
I	<b>58</b>	<b>50</b>	68	73	80	80	88	93
J	<b>52</b>	<b>42</b>	58	62	78	76	84	83
K	9	12	3	6	<b>21</b>	<b>27</b>	24	27
L	6	6	18	15	24	27	24	24
Avg.	49.06	49.06	57.20	60.54	70.46	70.04	75.68	76.83

For saving space, we only report the fusion result in Tab. 2. From the Tab. 2 it can be seen that 1) GEI-contour and CGI obtain a remarkable improvement on Exp. H, I, J compared with GEI, and CGI is slightly better than GEI-contour. It means the key to the improvement on briefcase condition is the contour. One possible reason is that contour weakens the influence from regions inside the briefcase’s silhouettes. 2) We also notice that CGI and GEI perform much more better than GEI-contour on Exp. D, E, F, G. It indicates that although contour instead of silhouette reduces the recognition rate on surface condition, using the proposed CGI temporal template can make up for such loss and further improve



**Table 4.** Comparison of Recognition Performance on USF HumanID Database using Different Methods, Rank1 Performance (%).

	A	B	C	D	E	F	G	H	I	J	K	L	Avg.
Baseline [11]	73	78	48	32	22	17	17	61	57	36	3	3	40.96
HMM [21]	89	88	68	35	28	15	21	<b>85</b>	<b>80</b>	58	17	15	53.54
IMED+LDA [14]	88	86	72	29	33	23	32	54	62	52	8	13	48.64
2DLDA [14]	89	<b>93</b>	80	28	33	17	19	74	71	49	16	16	50.98
DATER [13]	87	<b>93</b>	78	42	42	23	28	80	79	59	18	21	56.99
TLPP [22]	87	<b>93</b>	72	25	35	17	18	62	62	43	12	15	46.95
MTP [15]	90	91	<b>83</b>	37	43	23	25	56	59	59	9	6	51.57
GEI+Real [1]	89	87	78	36	38	20	28	62	59	59	3	6	51.04
GEI+Synthetic [1]	84	<b>93</b>	67	53	45	30	34	48	57	39	<b>21</b>	<b>24</b>	51.04
GEI+Fusion [1]	90	91	81	<b>56</b>	<b>64</b>	25	36	64	60	60	6	15	57.72
CGI+Real	90	89	81	28	30	15	13	82	75	60	3	3	51.98
CGI+Synthetic	89	89	67	53	53	27	33	59	60	56	3	15	54.49
CGI+Fusion	<b>92</b>	<b>93</b>	76	47	48	<b>34</b>	<b>40</b>	82	73	<b>62</b>	6	15	<b>60.54</b>

**Table 5.** Comparison of Recognition Performance on USF HumanID Database using Different Methods, Rank5 Performance (%).

	A	B	C	D	E	F	G	H	I	J	K	L	Avg.
Baseline [11]	88	93	78	66	55	42	38	85	78	62	12	15	64.54
HMM [21]	-	-	-	-	-	-	-	-	-	-	-	-	-
IMED+LDA [14]	95	95	90	52	63	42	47	86	86	78	21	19	68.60
2DLDA [14]	<b>97</b>	<b>93</b>	<b>93</b>	57	59	39	47	91	<b>94</b>	75	<b>37</b>	34	70.95
DATER [13]	96	<b>96</b>	<b>93</b>	69	69	51	52	92	90	83	40	36	75.68
TLPP [22]	94	94	87	52	55	35	42	85	78	68	24	33	65.18
MTP [15]	94	93	91	64	68	51	52	88	83	82	18	15	71.38
GEI+Real [1]	93	93	89	65	60	42	45	88	79	80	6	9	68.68
GEI+Synthetic [1]	93	<b>96</b>	<b>93</b>	75	71	54	53	78	82	64	33	<b>42</b>	72.13
GEI+Fusion [1]	94	94	<b>93</b>	<b>78</b>	<b>81</b>	<b>56</b>	53	90	83	82	27	21	76.30
CGI+Real	96	94	<b>93</b>	64	62	45	50	<b>94</b>	93	<b>85</b>	27	33	73.90
CGI+Synthetic	93	<b>96</b>	85	71	72	47	50	91	85	82	21	30	73.28
CGI+Fusion	96	94	<b>93</b>	75	70	54	<b>57</b>	<b>94</b>	93	83	27	24	<b>76.83</b>

the performance of gait recognition. 3) Compared with CGI-gray, CGI has better Rank1 performance in 9 out of 12 specific conditions and improve the average recognition ratio by about 5%. We can thus infer that with color encoding, the temporal information of the gait sequence benefits individual recognition by gait.

With the same parameter setting, we also investigate the influence of gait period detection as tabulated in Tab. 3. We observe from Tab. 3 that the divergence between the two detection methods is minor in almost all the experiments expect few groups of experiments highlighted in the table. One reason is that GEI, which uses the arithmetic average to generate the gait energy image, is

insensitive to key frame selection and period detection. At the same time, this experiment indicates that our method is robust to the period detection, it can work well using a basic period detection, and may work better if employing an advanced period detection method. Furthermore, CGI performs better than GEI when using both period detection methods.

Finally, we also compare the proposed algorithms with several state-of-the-art published results are illustrated as in Tab. 4 and Tab. 5. It is obvious that the proposed CGI outperforms others in both average Rank1 and Rank 5 performances, and is robust under most of the complex conditions. It is worth mentioning that since the time and space complexities of CGI are the same as those of GEI, the proposed CGI temporal template is very effective and competitive for real-world applications.

## 5 Conclusion

In this paper, we have proposed a simple and effective temporal template CGI. We extract a set of contour images from the corresponding silhouette images using local entropy principle, then color encoding the temporal information of gait sequence into the CGI. We also generate real and synthetic temporal templates and exploit fusion strategy to obtain better performance. Experiments in a benchmark database have demonstrated that compared with state-of-the-art, our CGI template can attain higher recognition accuracy.

In the future, we will study how to enhance CGI's robustness in more complex conditions, and investigate how to select a more general color mapping function instead of the current linear mapping function. Furthermore, we will consider to generalize the proposed frameworks into other human-movement-related fields [23] such as gesture analysis and abnormal behavior detection.

## Acknowledgements

This work was supported in part by the NFSC (No. 60635030, 60975044) and 973 program (No. 2010CB327900, 2009CB320903), Shanghai Leading Academic Discipline Project No. B114, Shanghai Key Laboratory of Intelligent Information Processing, China. Grant No. IPL-09-016, and Hui-Chun Chin and Tsung-Dao Lee Chinese Undergraduate Research Endowment (CURE).

## References

1. Han, J., Bhanu, B.: Individual Recognition using Gait Energy Image. TPAMI 28, 316–322 (2006)
2. Yam, C.Y., Nixon, M.S.: Model-based Gait Recognition. In: Encyclopedia of Biometrics, pp. 633–639. Springer, Heidelberg (2009)
3. Wang, L., Tan, T.N., Hu, W.M., Ning, H.Z.: Automatic Gait Recognition Based on Statistical Shape Analysis. TIP 12, 1120–1131 (2003)

4. Veres, G.V., Gordon, L., Carter, J.N., Nixon, M.S.: What Image Information is Important in Silhouette-based Gait Recognition? In: CVPR, vol. 2, pp. 776–782 (2004)
5. Guo, B., Nixon, M.S.: Gait Feature Subset Selection by Mutual Information. TSMC, Part A 39, 36–46 (2009)
6. Bouchrika, I., Nixon, M.S.: Model-based Feature Extraction for Gait Analysis and Recognition. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 150–160. Springer, Heidelberg (2007)
7. Wang, L., Tan, T., Ning, H., Hu, W.: Fusion of Static and Dynamic Body Biometrics for Gait Recognition. TCSVT 14, 149–158 (2004)
8. Wang, L., Tan, T.N., Ning, H.Z., Hu, W.M.: Silhouette Analysis Based Gait Recognition for Human Identification. TPAMI 12, 1505–1518 (2003)
9. Sundaresan, A., Roy-Chowdhury, A., Chellappa, R.: A Hidden Markov Model Based Framework for Recognition of Humans from Gait Sequences. In: ICIP, vol. 2, pp. 93–96 (2003)
10. Kobayashi, T., Otsu, N.: Action and Simultaneous Multiple-person Identification using Cubic Higher-order Local Auto-correlation. In: ICPR, vol. 4, pp. 741–744 (2004)
11. Sarkar, S., Phillips, P.J., Liu, Z., Vega, I.R., Grother, P., Bowyer, K.W.: The Humanid Gait Challenge Problem: Data Sets, Performance, and Analysis. TPAMI 27, 162–177 (2005)
12. Liu, Z., Sarkar, S.: Simplest Representation Yet for Gait Recognition: Averaged Silhouette. In: ICPR, vol. 4, pp. 211–214 (2004)
13. Xu, D., Yan, S., Tao, D., Zhang, L., Li, X., Zhang, H.J.: Human Gait Recognition With Matrix Representation. TCSVT 16, 896–903 (2006)
14. Tao, D., Li, X., Wu, X., Maybank, S.J.: General Tensor Discriminant Analysis and Gabor Features for Gait Recognition. TPAMI 29, 1700–1715 (2007)
15. Chen, C., Zhang, J., Fleischer, R.: Multilinear tensor-based non-parametric dimension reduction for gait recognition. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 1030–1039. Springer, Heidelberg (2009)
16. Bashir, K., Xiang, T., Gong, S.: Gait recognition using gait entropy image. IET Seminar Digests 2009 (2009)
17. Woodring, J., Shen, H.W.: Chronovolumes: A Direct Rendering Technique for Visualizing Time-varying Data. In: Eurographics, pp. 27–34 (2003)
18. Jänicke, H., Wiebel, A., Scheuermann, G., Kollmann, W.: Multifield Visualization using Local Statistical Complexity. TVCG 13, 1384–1391 (2007)
19. Wang, C., Yu, H., Ma, K.L.: Importance-driven Time-varying Data Visualization. TVCG 4, 1547–1554 (2008)
20. Yan, C., Sang, N., Zhang, T.: Local entropy-based transition region extraction and thresholding. Pattern Recognition Letters 24, 2935–2941 (2003)
21. Kale, A., Sundaresan, A., Rajagopalan, A.N., Cuntoor, N.P., RoyChowdhury, A.K., Kruger, V., Chellappa, R.: Identification of Humans using Gait. TIP 13, 1163–1173 (2004)
22. Chen, C., Zhang, J., Fleischer, R.: Distance Approximating Dimension Reduction of Riemannian Manifolds. TSMCB 40, 208–217 (2010)
23. Moeslund, T.B., Hilton, A., Krüger, V.: A Survey of Advances in Vision-based Human Motion Capture and Analysis. CVIU 103, 90–126 (2006)

# Robust Face Recognition Using Probabilistic Facial Trait Code

Ping-Han Lee<sup>1</sup>, Gee-Sern Hsu<sup>3</sup>, Szu-Wei Wu<sup>2</sup>, and Yi-Ping Hung<sup>2</sup>

<sup>1</sup> Department of Computer Science and Information Engineering, National Taiwan University

<sup>2</sup> Graduate Institute of Networking and Multimedia, National Taiwan University

<sup>3</sup> Department of Mechanical Engineering, National Taiwan University of Science and Technology

**Abstract.** Recently, Facial Trait Code (FTC) was proposed for solving face recognition, and was reported with promising recognition rates. However, several simplifications in the FTC encoding make it unable to handle the most rigorous face recognition scenario in which only one facial image per individual is available for enrollment in the gallery set and the probe set includes faces under variations caused by illumination, expression, pose or misalignment. In this study, we propose the Probabilistic Facial Trait Code (PFTC) with a novel encoding scheme and a probabilistic codeword distance measure. We also proposed the Pattern-Specific Subspace Learning (PSSL) scheme that encodes and recognizes faces robustly under aforementioned variations. The proposed PFTC was evaluated and compared with state-of-the-art algorithms, including the FTC, the algorithm using sparse representation, and the one using Local Binary Pattern. Our experimental study considered factors such as the number of enrollment allowed in the gallery, the variation among gallery or probe set, and reported results for both identification and verification problems. The proposed PFTC yielded significant better recognition rates in most of the scenarios than all the states-of-the-art algorithms evaluated in this study.

## 1 Introduction

Face recognition remains a popular topic in the recent two decades, and many algorithms were proposed. Ahonen et al. proposed using Local Binary Pattern (LBP) for face recognition [1]. In their algorithm, a face was spatially decomposed into several non-overlapping blocks. The histograms of the local binary patterns extracted from these blocks were concatenated to form a new feature vector. For the application to face recognition, the distance between two faces was evaluated using the *weighted Chi square distance* between their feature vectors. The LBP approach has been reported to be quite robust to the facial variations caused by illumination and expression changes [1]. A face recognition algorithm using Sparse Representation proposed by Wright et al. [2] has aroused some attention recently. They treated a test sample as a sparse linear

combination of training samples, and computed this sparse representation using  $l^1$ -minimization. The face recognition algorithm proposed in [2], denoted as **SRC**, yielded robust results when faces were occluded or corrupted. It is worthy to mention that both algorithms, LBP and SRC, do not involve a training stage that learns knowledge specific to human faces.

In contrast to LBP and SRC, the Facial Trait Code (**FTC**) proposed by Lee et al. learned some face-specific knowledge. It has been shown in [3] that patterns exist in many of the facial components and can be extracted, and the extracted patterns can be used to decompose and encode a face. The facial components with patterns good for discriminating faces were called *facial traits*, and the associated face coding scheme was called the *Facial Trait Code*. It is reported the patterns learned from a small set of human faces can be generalized to human face unseen in the training stage, and FTC yielded comparable face identification rates with the LBP approach, and significantly better verification rates [3].

Although the three algorithms were reported to be successful, they are limited in some aspects. The SRC approach assumed any probe, an image in the probe set<sup>1</sup>, could be represented as a linear combination of the gallery images. It is not known whether SRC performs well when there is only few, or even one image per individual allowed for enrollment in gallery. Like SRC, the LBP approach does not learn any illumination- or expression-invariants among human faces. If the probes were taken under conditions (i.e. illumination or facial expression) significantly different from those the images in gallery were taken, the performance of the LBP approach is expected to degrade noticeably. FTC does not require many samples per individual for enrollment, and it learns a way to encoding faces robustly under illumination variation. However, it suffers from the following aspects:

1. Because the distance measure in the FTC code space is given by Hamming distance, the similarities between one facial trait patch and all facial trait patterns except the most similar one are ignored.
2. Because the impacts from illumination, expression, and pose<sup>2</sup> variations upon the facial trait patterns have not been studied thoroughly, a systematic way to encompass these impacts into the FTC has yet to develop.
3. Because of the above and a few simplifications, the FTC cannot handle the most rigorous face recognition scenario in which only one facial image per individual is available for enrollment in the gallery set and the probe set includes faces under variations in illumination, expression and pose (e.g. John has a smiling, uniformly-lit face enrolled in the database, and we want to recognize John's face which is left-lit and with neutral expression).

<sup>1</sup> According to the FRVT 2006[4], the target set contains facial images for enrollment to the face recognition system, and the probe set contains images to be recognized. If only one image per individual is allowed for enrollment, then the target set is called the gallery set. In this paper, we do not distinguish between the target set and the gallery, and we will use the term *gallery* in the rest of this paper.

<sup>2</sup> Or, out-of-plane rotation. We will use the two terms interchangeably in the rest of this paper.

In this paper, we propose the *Probabilistic Facial Trait Code* (PFTC) that solves all of the above issues. A comprehensive performance evaluation of some state-of-the-art algorithms as well as the proposed PFTC are also given. All the aforementioned algorithms, LBP, SRC and FTC, were evaluated under several different scenarios, which differ in the number of enrollment allowed in the gallery set, the variations among the gallery and the probe set, and the definitions of training set, gallery and probe set. The results in AR dataset [5] in which faces were taken under *controlled* variations, as well as a mixed dataset in which faces were taken *uncontrolled* conditions were given. The proposed PFTC yielded significantly better recognition rates than the three recent face recognition algorithms in most of the scenarios considered in this paper.

This paper begins with an introduction to Facial Trait Code (Sec. 2). The development of the Probabilistic Facial Trait Code was given in Sec. 3. A comparative study on the face recognition performance using the PFTC and other algorithms were reported in Sec. 4. The conclusion and contribution of our study were summarized in Sec. 5.

## 2 Introduction to Facial Trait Code

Two face datasets are needed for the construction of FTC, one is the *Trait Extraction Set*, denoted as **TES**, and the other the *Trait Variation Set*, denoted as **TVS**. The former consists of a large number of frontal facial images with neutral expression and evenly distributed illumination, and is used to determine the facial traits and the patterns in each trait (Sec. 2.1). The latter consists of facial images taken under various illumination conditions, and is used as an add-on to the facial trait samples so that each trait pattern can have samples with illumination changes (Sec. 2.2). For the Facial Trait Code and the proposed Probabilistic FTC, samples in **TES** were used to extract *patch patterns* and to select *facial traits*. When facial traits and associated patterns were determined, samples in both **TVS** and **TES** were used to train trait-specific SVMs.

### 2.1 Facial Trait Extraction and Associated Codewords

A local patch on a face can be specified by a rectangle bounding box  $\{x, y, w, h\}$ , where  $x$  and  $y$  are the 2-D pixel coordinates of the bounding box's upper-left corner, and  $w$  and  $h$  are the width and height of this bounding box, respectively. A large amount of patches with different sizes and locations on faces can be defined, and slightly more than a couple thousands of patches for a face with 80x100 pixels in size are used in [3]. In the following, we assume  $M$  patches in total obtained from a face.

Assuming  $K$  faces available from the TES, and all faces aligned by the centers of both eyes, we will obtain a stack of  $K$  patch samples in each patch. To cluster the  $K$  patch samples in each patch stack, the Principal Component Analysis (PCA) followed by the Linear Discriminant Analysis (LDA) [6] are applied

to extract the features. It is assumed that these low dimensional patch features in each patch stack can be modeled by a Mixture of Gaussian (MoG), then the unsupervised clustering algorithm proposed by Figueiredo and Jain [7] can be applied to identify the MoG patterns in each patch stack. Assuming  $M$  patch stacks are available, this algorithm can cluster the low dimensional patch features into  $k_i$  clusters in the  $i$ -th patch stack, where  $i = 1, 2, \dots, M$ . The  $k_i$  clusters in the  $i$ -th patch stack were considered the patterns existing in this patch stack, and they are called the **patch patterns**.

A scheme was proposed in [3] that selects some combination of the patches with their patch patterns able to best discriminate the individuals in the TES by their faces. This scheme first define a matrix, called **Patch Pattern Map (PPM)**, for each patch.  $PPM$  shows which individuals' faces reveal the same pattern at that specific patch. Let  $PPM_i$  denote the  $PPM$  for the  $i$ -th patch,  $i = 1, 2, \dots, M$ .  $PPM_i$  will be  $L \times L$  in dimension in the case with  $L$  individuals, and the entry at  $(p, q)$ , denoted as  $PPM_i(p, q)$ , is defined as follows:  $PPM_i(p, q) = 0$  if the patches on the faces of the  $p$ -th and the  $q$ -th individuals are clustered into the same patch pattern and  $PPM_i(p, q) = 1$  otherwise.

Given  $N$  patches and their associated  $PPM_i$ 's stacked to form a  $L \times L \times N$  dimensional array, there are  $L(L-1)/2$   $N$ -dimensional binary vectors along the *depth* of this array because each  $PPM_i$  is symmetric matrix and one can only consider the lower triangular part of it. Let  $v_{p,q}$  ( $1 \leq q < p \leq L$ ) denote one of the  $N$ -dimensional binary vectors, then  $v_{p,q}$  reveals the local similarity between the  $p$ -th and the  $q$ -th individuals in terms of these  $N$  local patches. More unities in  $v_{p,q}$  indicates more differences between this pair of individuals, and on the contrary, more zeros shows more similarities in between.

The binary vector  $v_{p,q}$  motivated the authors in [3] to apply the Error Correcting Output Code (ECOC) [8] to their study. If each individual's face is encoded using the most discriminant patches, then the induced set of  $[v_{p,q}]_{1 \leq q < p \leq L}$  can be used to define the minimum and maximum Hamming distance [9] among all encoded faces in the corresponding code space. The  $v_{p,q}$  with the least (most) of unities gives the minimum (maximum) Hamming distance. To maximize the robustness against possible recognition errors in the decoding phase, authors in [3] proposed an Adaboost algorithm to maximize the  $d_{min}$ , the minimum Hamming distance, for the determination of the most discriminating from the overall patches[3]. Patches that best discriminate faces of different individuals were called the **facial traits**, and the associated patch patterns were dubbed as the **distinctive trait patterns**.

Assuming  $N$  facial traits selected from the the overall  $M$  patches, and each has  $k_j$ ,  $j = 1, 2, \dots, N$ , trait patterns, one can now define the codewords in FTC. Each codeword is of length  $N$  and  $n$ -ary where  $n$  is the largest number of the trait patterns found in one single trait, and each digit in a codeword is an integer number indicating a trait pattern. In summary, given a large collection of faces as the TES, one can define  $N$  facial traits,  $\sum_{j=1}^N k_j$  trait patterns, and  $\prod_{j=1}^N k_j$  faces (or FTC codewords).

---

<sup>3</sup> Due to the page limit, please refer to [3] for the details on this Adaboost algorithm.

## 2.2 FTC Encoding and Decoding

To apply the FTC to face recognition, the images in a gallery set are firstly encoded into **gallery codes** using a trait-specific SVM (Support Vector Machines) classifier [10] able to classify each facial trait into a symbolized trait pattern. This SVM classifier can be made using the trait samples from both the TES and the TVS for encompassing possible variations in each trait. In the decoding phase when a probe, an image from a probe set, is given, it is also firstly encoded into a **probe code**, and then matched against the gallery codes using Hamming distance as the measure. Given two codewords, one is a gallery code  $\mathbf{g}_c = [g_1 g_2 \dots g_N]$  and the other probe code  $\mathbf{p}_c = [p_1 p_2 \dots p_N]$ , the Hamming distance is give by the code difference  $\mathbf{d}_c = [d_1 d_2 \dots d_N]$  where  $d_i = 0$  if  $p_i = g_i$ , and  $d_i = 1$  otherwise. Then the Hamming distance between  $\mathbf{g}_c$  and  $\mathbf{p}_c$  is given by  $D(\mathbf{g}_c, \mathbf{p}_c) = \sum_{i=1}^N d_i$ .

## 3 Probabilistic Facial Trait Code

FTC suffers from the following aspects: the simplified integer codewords ignore the similarities between one facial trait patch and all facial trait patterns; a systematic way to handle the impacts from illumination, expression, or pose is yet to be developed for robust encoding. In this paper, we proposed the Probabilistic Facial Trait Code, or **PFTC**, that considers and solves the above issues. The primary differences from the original FTC include the following:

1. Instead of using only one integer to denote a facial trait in the encoding phase, we allow the probabilities of the trait belonging to all trait patterns in the codeword. The gain, to be shown in our experimental study, is the superb accuracy, on the price of more memory occupied by such a codeword. This leads to a complete revision of the distance measure between codewords and also the encoding and decoding schemes, as will be described in Sec. 3.1.
2. In addition to the maximization of the discrimination of the faces in the TES, we also maximize the discrimination of the trait patterns in each trait, based on a new TVS that includes variations caused by illumination, expression, and pose, as well as a new *Trait Enrichment Set (TRS)* that includes imprecisely aligned trait patches. This scheme makes the proposed PFTC robust in recognizing faces under aforementioned variations, and it will be described in 3.2.

### 3.1 Probabilistic Encoding and Decoding

Consider a  $N$ -trait FTC codeword  $\mathbf{g} = [g_1 g_2 \dots g_N]$ , where each  $g_i$  is an integer representing the  $g_i$ -th pattern that the  $i$ -th trait sample belongs to. It will take the form  $\mathbf{G} = [G_1 G_2 \dots G_N]$  in the PFTC, where each  $G_i$  is a vector with each element measuring the probability of this trait sample to one specific pattern of that trait. The dimension of  $G_i$  is the same as the number of the patterns extracted from the TES for that trait. We measure the probability of a trait



**Table 1.** Comparison between the hard and probabilistic codewords encoded by  $N$  facial traits

codeword type	hard	probabilistic
data structure	$N$ integers	$N$ $k$ -by-1 real arrays (typically, $k < 100$ )
encoding complexity	$N$ SVM classifications	
decoding complexity	$N$ integer comparisons	$N$ Bhattacharyya distance calculations

sample belonging to a trait pattern using its distance to the support hyperplane given by the SVM classifier of that trait pattern. To make  $G_i$  a distribution, we normalize its magnitude so that  $\|G_i\| = 1$ . The distance between two PFTC codewords,  $\mathbf{G}_a = [G_{a,1} G_{a,2} \dots G_{a,N}]$  and  $\mathbf{G}_b = [G_{b,1} G_{b,2} \dots G_{b,N}]$ , is defined as the following:

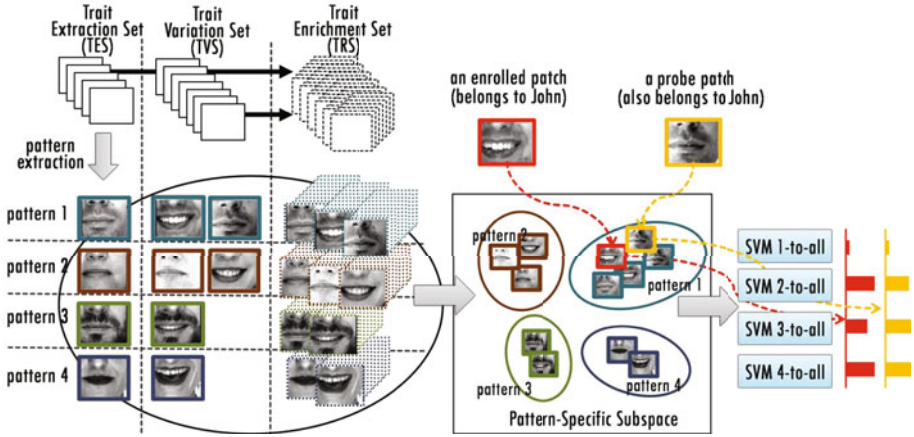
$$D(\mathbf{G}_a, \mathbf{G}_b) = \sum_{i=1}^N B(\mathbf{G}_{a,i}, \mathbf{G}_{b,i}) \quad (1)$$

where  $B(G_{a,i}, G_{b,i})$  is the *Bhattacharyya distance* [?] between the two pattern distributions  $G_{a,i}$  and  $G_{b,i}$  at the  $i$ -th trait.

We denote the integer codewords described in [3] as the **hard codewords**, and denote the proposed ones as the **probabilistic codewords**. The recognition results using both types of codewords will be reported and compared in our experimental study. The probabilistic codewords require a bit more storage space and computation than the hard ones, in exchange of superior recognition accuracy. TABLE 1 gives a comparison between the hard and the probabilistic codewords. Consider a FTC with 64 facial traits, and each facial trait has at most 64 trait patterns. A corresponding hard codeword requires only 48 Bytes for storage, while a probabilistic codewords requires 16 KBytes, when *single precision* numbers are used.

### 3.2 Trait Enrichment Set and the Pattern-Specific Subspace Learning

Similar to FTC, for the application of the PFTC to face recognition problem, a given face is encoded into a codeword, and matched, or decoded, into one of the gallery codeword. Ideally, if two faces belong to the same individual, they should be encoded exactly the same. However, in practice the facial variations in illumination conditions, expressions, poses or even the misalignments of facial images pose challenges for robust encoding. To recognize face correctly under a very strict scenario where only one facial image is allowed for enrollment and probe images are under aforementioned variations (e.g. John has a smiling, uniformly-lit face enrolled in the database, and we want to recognize John's face which is left-lit and with neutral expression), PFTC is required to encode faces of the same individual *similarly*, no matter under what conditions these facial images are taken.



**Fig. 1.** Illustration of the pattern-specific subspace learning scheme. In this illustration, we extract four patterns for this trait using the facial images in TES. With patches in TVS and TRS added, we learn the pattern-specific subspace where patches of different patterns are well separated. Given a patch from an enrolled face, it is projected into the pattern-specific subspace, and then its distance to the four one-to-the-rest SVMs are calculated and concatenated to form the trait-specific probabilistic distribution. The associated distribution of a probe patch is also calculated and compared with the enrolled distribution.

The *Pattern-Specific Subspace Learning* scheme, denoted as **PSSL**, proposed in this paper is the solution to this requirement. We learn subspaces in which trait patterns can be best discriminated and impacts of variations in illuminations, expressions, poses and misalignments are minimized. The proposed PFTC using the PSSL scheme is illustrated in Fig. 1, and it is composed of the following steps:

1. We collect a large set of faces taken under variations caused by illumination, facial expression and pose and add them into the **TVS**<sup>4</sup>
2. The facial trait pattern extraction is the same as that in the [3]: extraction of patch patterns using the clustering approach given in [7] and then the Adaboost selection scheme upon the *PPMs* using the faces from the **TES**.
3. All facial images in this work were aligned by centers of eyes, and thus the localization errors of eyes lead to the misalignment of facial images in practice. To handle this problem, for each image in the TES and TVS, we allowed the center of one eye to have 2 pixel offset from the *true center* in each direction, up, down, right, left. This gave 5 possible eye centers for an eye, and both eyes resulted in 25 possible eye-center pairs. We cropped

<sup>4</sup> The **TVS** in [3] contained only facial images taken under minor illumination variations.

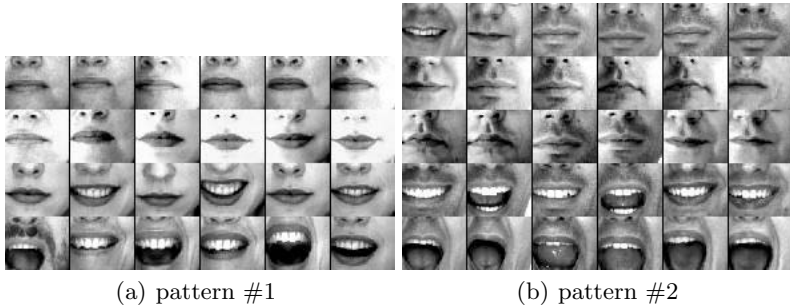
the face using these 25 pairs, left the one with true centers in the TES or TVS, and moved the rest 24 to a new set defined in this paper, the *Trait Enrichment Set (TRS)*<sup>5</sup>

4. Since the individuals in the **TVS** and **TRS** must be in **TES**, each *variational* facial trait sample from the **TVS** and **TRS** must have at least one corresponding *neural* trait sample<sup>6</sup> in **TES**. We merged the neural trait samples and the variation trait samples in each trait, and then applied the Linear Discriminant Analysis (LDA) to extract the features that maximize the scattering between different *trait patterns* while minimize the scattering within each trait pattern. This step forced each trait feature to include the trait's variations caused by illumination, expression, pose, and misalignment.
5. For each trait pattern in a facial trait, a SVM classifier was trained using the *one-to-the-rest* scheme with the LDA features extracted in Step-4. This gave the same number of SVM classifiers as that of the patterns existing in the facial trait.
6. In the encoding phase, a gallery face was firstly decomposed into a set of facial traits, and each trait was encoded using its normalized distances to the hyperplanes of all SVM classifiers for that specific trait. This step converts a gallery face into a gallery codeword.
7. In the decoding phase, a probe face was also converted into a probe codeword in the same manner, and then the distance measure given by (II) was used to determine if it matched any gallery codeword.

As shown in Fig. II, assume we have an enrolled patch and a probe patch both belong to John, and John's patch belongs to the first pattern. The enrolled patch is taken under uniform lighting with mouth smiling, and it is the only enrolled patch for John. Although we do not know the actual appearance of a left-lit patch with neutral expression of John, as long as we have a training individual whose patches are under the same condition and also belong to the first pattern, we can recognize John's probe patch shown in Fig. II through the proposed PSSL scheme. This scheme exploits an observation on human facial images: *if the neutral patches (i.e. uniformly-lit patches with neutral expressions) of two individuals are classified into the same pattern, the patches of the two individuals under the same illumination condition will have similar appearance, so will those make the same facial expression.* Fig. II illustrates this observation. The proposed PSSL scheme grants the PFTC the ability to handle the most strict scenario where only one facial image is allowed for enrollment and probe images are under aforementioned variations, and it will be validated in our experimental study in Sec. IV.

<sup>5</sup> Similarly, one can tolerate larger degree of misalignment of the eyes by allowing the center of one eye to have more than two pixel offset.

<sup>6</sup> An evenly lighted frontal face without facial expression.



**Fig. 2.** Illustration of patches of two patterns under different illumination conditions and facial expressions. (a) Patches belong to three different individuals that are classified into pattern #1. The first row: neutral patches; the second row: patches under a similar illumination condition; the third and the fourth row: patches make two facial different expressions. (b) Patches belong to three different individuals that are classified into pattern #2. The first row: neutral patches; the second and the third row: patches under two different illumination conditions; the fourth and the fifth row: patches make two different facial expressions.

## 4 Performance Evaluation—A Comparative Study

The training set, the gallery (or target) set, and the probe set are generally three disjoint sets. The training set is composed of the TES, TVS and TRS datasets for building up the facial traits and their associated trait patterns. If the gallery set happens to be the training set, i.e., the trait patterns are all learned from the gallery set, the performance of the FTC is expected to reach its best. It would be interesting to study the difference in performance between this best case and the general case that the trait patterns are already defined from the training set, and the gallery set can only be encoded using the training-set defined trait patterns. Therefore, two test protocols are considered as follows:

- **Protocol-1:** the training set and gallery set are the same;
- **Protocol-2:** the training set and gallery set are two disjoint sets.

Both protocols were tested on the AR face database [5] and a dataset composed of samples from several face databases. The tests on the AR database gave a way to compare the FTC against other algorithms with reported performance on AR database. However, to reveal that the facial traits can be better defined from a large set of faces collected from different resources, and compared with other algorithms in performance, a *mixed* dataset was used. The mixed dataset is composed of images collected from AR [5], FERET [12], FRGC [13], FVI [16], PIE [14] and XM2VTS [15]. The mixed dataset includes 6405 facial images from 903 individuals taken under uncontrolled illumination conditions, facial expressions, and poses, or using different image acquisition devices. The pose variations cover at most  $20^\circ$  toward both sides, up and down. The details of the samples selected

**Table 2.** The mixed dataset: the details of the samples selected from the six datasets

Dataset	#subjects	#faces	#face/subject	note
AR [5]	126	1764	14	all facial images without occlusion such as sunglass and scarf
FERET [12]	200	400	2	ba(neutral) and bk(with illumination variation)
FRGC [13]	201	1957	1 ~ 25	1957 images with their illumination conditions, facial expressions and poses manually annotated
FVI	38	760	20	10 images from one session and 10 from another session. Images in the two sessions were collected two weeks apart.
PIE [14]	43	344	8	random 8 images for each person taken under room lights on with flash lights of different directions
XM2VTS [15]	295	1180	4	the speech shot
total	903	6405	7.09	

from the six datasets are given in TABLE 2. All facial images in both datasets were aligned to the centers of the eyes, and normalized to 80x100 pixels in size.

Two typical face recognition tasks were carried out: identification and verification. In identification, each probe image had one unique match to identify in the gallery set. In verification, each probe image with a claimed subject were both presented to the verification algorithm, which would either accept or reject the claim. A claim would be rejected when the probe failed to match the claimed subject, no matter whether the subject of the probe was in the gallery set or not.

**Table 3.** The test protocols. 'E', 'P', 'D', and 'SPE' stands for 'Experiment', 'Protocol', 'Dataset', and 'the number of Sample-Per-Enrollee' respectively. 'S' and 'F' stands for 'Subject' and 'Face' respectively.

E	P	D	training	gallery	probe		SPE	descriptions of the data partitions
					enrollee	imposter		
1	1	AR	$A_1$ 126S, 882F		$A_2$ 126S, 882F	-	7	$A_1$ and $A_2$ , two disjoint sets from AR, parted by different shooting time.
2			$B_1$ 63S 882F	$B_{2a}$ 63S 63F~441F	$B_{2b}$ 63S 441F	-	1~7	$B_1$ and $B_2$ , two disjoint sets from AR, parted by different subject. $B_2$ further divided into $B_{2a}$ and $B_{2b}$ , parted by different shooting time.
3	2	mixed	$C_1$ 304S 2630F	$C_{2a}$ 303S 303F	$C_{2b}$ 303S 975F	$C_3$ 296S 1950F	1	$C_1$ , $C_2$ and $C_3$ , three disjoint sets from the mixed dataset, parted by different subject. $C_2$ further divided into $C_{2a}$ and $C_{2b}$ , parted by different shooting time.

We compared the FTC's performance with LBP [1] and the algorithm using Sparse Representation (SRC) [2]. SRC is acknowledged as one of the most potential approaches for face recognition published recently. Our comparison also included on algorithm using local patches [17], one algorithm using ECOC [18], as well as two baseline methods, Eigenface [19] and Fisherface [20]. The early version of FTC [3] with hard codewords was also included in this comparison

**Table 4.** Results of Experiment 1 (in percentage). The 'IDT' and 'EER' stand for 'identification rate' and 'equal-error rate measured when FAR equals to FRR', respectively.

algorithm	IDT	HIT at FAR equals to			EER
		$10^{-1}$	$10^{-2}$	$10^{-3}$	
<i>Eigenface</i> [19]	77.8	90.1	67.8	48.5	10.2
<i>Fisherface</i> [20]	80.5	89.7	69.8	52.7	10.0
<i>Heisele03</i> [17]	82.2	93.7	87.3	78.1	9.82
<i>ECOC</i> [18]	83.3	92.5	88.9	82.8	7.26
<i>LBP</i> [1]	92.4	93.0	78.6	57.8	8.31
<i>SRC</i> [2]	90.5	99.7	96.9	91.0	2.0
<i>FTC</i> [3]	<b>96.8</b>	<b>99.8</b>	<b>99.4</b>	<b>97.2</b>	<b>1.1</b>
<i>PFTC</i>	95.0	98.9	96.2	85.6	2.4

and dubbed as *FTC*. The proposed *PFTC* using both *PSSL* scheme and probabilistic codewords is dubbed as *PFTC*. For the *SRC* algorithm, we implemented the **Eigen+SRC** described in [2], which applied the PCA features. The implementation of *ECOC* applied [127, 15, 27] binary BCH code to generate the codewords<sup>7</sup>.

TABLE 3 summarizes the three experiments performed in this study. The *training set* is the set of images for training algorithms: it was used to extract the eigen-component in *Eigenface*, *Fisherface*, and *SRC*; it was divided into **TES** and **TVS** in *FTC*-based algorithms. The *probe* set contains faces for testing. If a *probe*, a face in the probe set, also exists in the gallery, then it is known as an *enrollee*; otherwise, it will be referred to as an *imposter*.

Experiment 1 studied the performance under Protocol-1, where the gallery set is the same as the training set. Experiment 2 studied performance variations with *SPE* (number of Sample-Per-Enrollee). Experiment 3 studied the performance variations caused by illumination, expression and pose under most challenging scenario where  $SPE = 1$ . The experimental results and discussions are listed below:

- TABLE 4 gives the results of Experiment 1. Both *FTC* and *PFTC* outperformed most algorithms, except for *SRC*. *FTC* and *PFTC* outperformed *SRC* in identification rate, but *SRC* appeared slightly better than *PFTC* in hit rate and ERR. However, *FTC* appeared to give the best overall performance.
- The identification and hit rates for Experiment 2 are shown in Fig. 3 (a) and (b), respectively. The performance of most algorithms degraded significantly when *SPE* decreased, except *FTC* and *PFTC*. *PFTC* (the red dotted lines) gives the best overall performance in this test.
- TABLE 5 summarizes the results of Experiment 3, in terms of identification and verification rates. With the most challenging scenario considered, where

<sup>7</sup> According to our unpublished results, *ECOC* did not achieve good recognition rates when using codewords less than 127 bits.

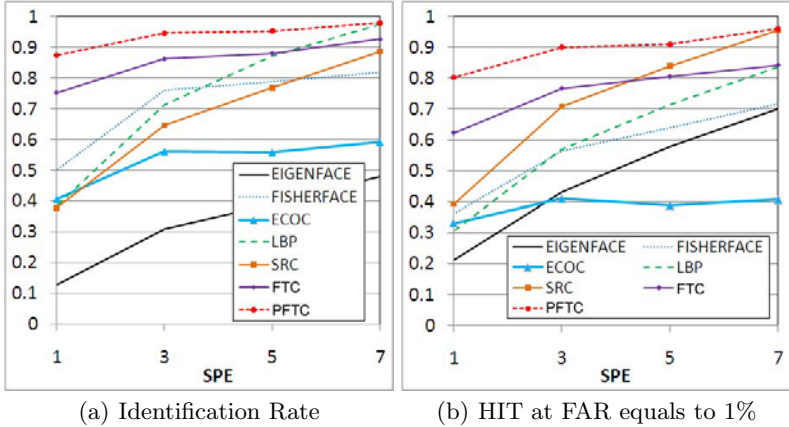


Fig. 3. Recognition results for Experiment 2

Table 5. Results of Experiment 3 (in percentage). The 'IDT' stands for the identification rate; 'HIT' is the hit rate measured at FAR=0.01; 'EER' stands for the equal-error rate, the rate at which both accept and reject errors are equal. The **neutral set** includes frontal faces with no variation; the **illumination, expression and pose set** includes faces taken under variation in illumination, expressions and pose, respectively.

subset	neutral set		variant set												
			illumination set			expression set			pose set			variant overall			
#faces	1806		490			410			219			1119			
algorithm	IDT	verification		IDT	verification		IDT	verification		IDT	verification		IDT	verification	
		HIT	EER		HIT	EER		HIT	EER		HIT	EER		HIT	EER
<i>Eigenface</i> [19]	54.8	51.2	21.9	19.4	14.3	30.0	47.3	34.2	22.8	52.1	54.6	21.6	36.0	29.5	25.7
<i>Fisherface</i> [20]	68.5	60.9	15.1	60.4	49.0	14.3	57.8	49.2	16.8	63.5	59.1	13.6	60.1	51.0	15.1
<i>ECOC</i> [18]	50.5	66.2	13.9	39.6	55.8	14.7	43.2	60.5	17.2	35.4	58.3	14.8	40.1	58.0	15.6
<i>LBP</i> [1]	77.7	62.1	15.1	60.4	56.1	20.2	<b>72.6</b>	65.6	16.5	59.4	44.1	27.6	64.7	57.2	20.3
<i>SRC</i> [2]	70.1	76.3	13.8	21.5	29.4	33.1	47.4	52.8	22.3	66.7	69.5	16.2	39.8	45.8	25.8
<i>FTC</i> [3]	80.9	85.1	6.89	68.1	77.9	7.91	60.0	66.3	9.99	58.3	74.8	8.61	63.2	73.0	8.81
<i>PFTC</i>	<b>87.8</b>	<b>88.0</b>	<b>4.83</b>	<b>84.0</b>	<b>85.0</b>	<b>5.22</b>	68.4	<b>67.1</b>	<b>8.34</b>	<b>83.3</b>	<b>86.3</b>	<b>6.51</b>	<b>78.1</b>	<b>78.7</b>	<b>6.62</b>

each enrollee has only one facial sample for enrollment (SPE=1), *PFTC* gave the best overall performance. It outperformed *FTC*, *SRC* and *LBP* by 14.9% 38.3% 13.4% respectively in identification rate. Similar performance was observed in hit rates and ERRs as well.

## 5 Conclusions and Future Works

In this paper, we proposed the Probabilistic Facial Trait Code to handle the most rigorous face recognition scenario with one gallery face per individual and probe faces under variations caused by illumination, expression and pose. The *PFTC* comes with novel encoding scheme, and it gives a much better performance than its predecessor, the [3]. Instead of using only one integer to denote a facial trait

in the encoding phase, PFTC allows the probabilities of the trait belonging to all trait patterns in the codeword. Furthermore, we included facial images under variations caused by illumination, expression, pose and misalignment, and learnt a pattern-specific subspace, which makes the proposed PFTC robust in recognizing faces under aforementioned variations.

The extensive experimental study given in this paper also evaluated three recent face recognition algorithms, one based on local binary pattern, one using the sparse representation, and the Facial Trait Code approach, under several scenarios with different definitions of training set, gallery and probe set and different conditions facial images were taken. The proposed PFTC outperformed all the state-of-the-art algorithms compared in this paper, especially when there is only one single image per individual allowed in the gallery.

## Acknowledgement

This work was supported in part by the Ministry of Economic Affairs, Taiwan, under Grant 98-EC-17-A- 02-S1-032 and by the National Science Council, Taiwan, under grant NSC 98-2221-E-002-127-MY3.

## References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2037–2041 (2006)
2. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 210–227 (2009)
3. Lee, P.H., Hsu, G.S., Hung, Y.P.: Face verification and identification using facial trait code. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1613–1620 (2009)
4. Phillips, J.P., Scruggs, T.W., O’toole, A.J., Flynn, P.J., Bowyer, K.W., Schott, C.L., Sharpe, M.: FRVT 2006 and ICE 2006 large-scale results. Technical report, National Institute of Standards and Technology (2007)
5. Martinez, A., Benavente, R.: The ar face database. Technical Report 24, CVC (1998)
6. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley, Chichester (2000)
7. Figueiredo, M., Jain, A.: unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 381–396 (2002)
8. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via errorcorrecting output codes. *Journal of Artificial Intelligence Research* 2, 263–286 (1995)
9. Lin, S., Costello, D.J.: *Error Control Coding*, 2nd edn. Pearson Education International, London (2004)
10. Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge (2000)



11. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society* 35, 99–109 (1943)
12. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1009–1034 (2000)
13. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 947–954 (2005)
14. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) database of human faces. Technical Report CMU-RI-TR-01-02, Carnegie Mellon University (2001)
15. Messer, K., Matas, J., Kittler, J., Luetten, J., Maitre, G.: Xm2vtsdb: The extended m2vts database. In: *Proceedings of the Second International Conference on Audio and Video-based Biometric Person Authentication* (1999)
16. Lee, P.H., Chu, L.J., Hung, Y.P., Shih, S.W., Chen, C.S., Wang, H.M.: Cascading multimodal verification using face, voice and iris information. In: *IEEE International Conference on Multimedia and Expo.*, Beijing, China, pp. 847–850 (2007)
17. Heisele, B., Ho, P., Wu, J., Poggio, T.: Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding* 91, 6–12 (2003)
18. Kittler, J., Ghaderi, R., Windeatt, T., Matas, J.: Face verification using error correcting output codes. In: *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I-755–I-760 (2001)
19. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3, 71–86 (1991)
20. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 711–720 (1997)

# A 2D Human Body Model Dressed in Eigen Clothing

Peng Guan<sup>1</sup>, Oren Freifeld<sup>2</sup>, and Michael J. Black<sup>1</sup>

<sup>1</sup> Department of Computer Science

<sup>2</sup> Division of Applied Mathematics

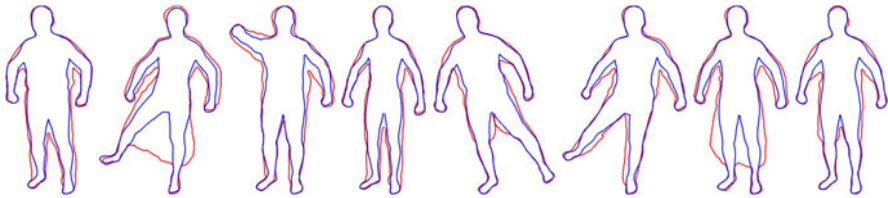
Brown University, Providence, RI 02912, USA

{pguan,black}@cs.brown.edu, freifeld@dam.brown.edu

**Abstract.** Detection, tracking, segmentation and pose estimation of people in monocular images are widely studied. Two-dimensional models of the human body are extensively used, however, they are typically fairly crude, representing the body either as a rough outline or in terms of articulated geometric primitives. We describe a new 2D model of the human body contour that combines an underlying naked body with a low-dimensional clothing model. The naked body is represented as a Contour Person that can take on a wide variety of poses and body shapes. Clothing is represented as a deformation from the underlying body contour. This deformation is learned from training examples using principal component analysis to produce *eigen clothing*. We find that the statistics of clothing deformations are skewed and we model the *a priori* probability of these deformations using a Beta distribution. The resulting generative model captures realistic human forms in monocular images and is used to infer 2D body shape and pose under clothing. We also use the coefficients of the eigen clothing to recognize different categories of clothing on dressed people. The method is evaluated quantitatively on synthetic and real images and achieves better accuracy than previous methods for estimating body shape under clothing.

## 1 Introduction

Two-dimensional models of the human body are widely used in computer vision tasks such as pose estimation, segmentation, pedestrian detection and tracking. Such 2D models offer representational and computational simplicity and are often preferred over 3D models for applications involving monocular images and video. These models typically represent the shape of the human body coarsely, for example as a collection of articulated rectangular patches [1,2,3,4]. None of these methods explicitly models how clothing influences human shape. Here we propose a new fully generative 2D model that decomposes human body shape into two components: 1) the shape of the naked body and 2) the shape of clothing relative to the underlying body. The naked body shape is represented by a 2D articulated Contour Person (CP) [5] model that is learned from examples. The CP model realistically represents the human form but does not model clothing.



**Fig. 1. Samples from the Dressed Contour Person model.** Different body shapes and poses (blue) are dressed in different types of eigen clothing (red).

Given training examples of people in clothing with known 2D body shape, we compute how clothing deviates from the naked body to learn a low-dimensional model of this deformation. We call the resulting generative model the *Dressed Contour Person* (DCP) and samples from this model are shown in Fig. 1.

The DCP model can be used just like previous models for person detection, tracking, etc. yet it has several benefits. The key idea is to separate the modeling of the underlying body from its clothed appearance. By explicitly modeling clothing we infer the most likely naked body shape from images of clothed people. We also solve for the pose of the underlying body, which is useful for applications in human motion understanding. The learned model accurately captures the contours of clothed people making it more appropriate for tracking and segmentation. Finally, the model supports new applications such as the recognition of different types of clothing from images of dressed people.

There are several novel properties of the DCP model. First we define *eigen clothing* to model deformation from an underlying 2D body contour. Given training samples of clothed body contours, where the naked shape of the person is known, we align the naked contour with the clothing contour to compute the deformation. The eigen-clothing model is learned using principal component analysis (PCA) applied to these deformations. A given CP model is then “clothed” by defining a set of linear coefficients that produce a deformation from the naked contour. This is illustrated in Fig. 1.

There is one problem, however, with this approach. As others have noted, clothing generally makes the body larger [6,7]. A standard eigen-model of clothing could generate “negative clothing” by varying the linear coefficients outside the range of the training samples. While non-negative matrix factorization could be used to learn the clothing model, we show that a simple prior on the eigen coefficients addresses the issue. In particular, we show that the eigen coefficients describing clothing deformations are not Gaussian and we model them using Beta distributions that capture their asymmetric nature.

We also demonstrate the estimation of a person’s 2D body shape under clothing from a single image. Previous work on estimating body shape under clothing has either used multiple images [6] or laser range scan data [7]. These previous approaches also did not actually model clothing but rather tried to ignore it. Both of the above methods try to fit a naked body that lies inside the measurements (images or range scans) while strongly penalizing shapes that are “larger”

than the observations. We show that there is a real advantage to a principled statistical model of clothing. Specifically we show accuracy in estimating naked body shape that exceeds that of Bălan and Black [6], while only using one uncalibrated image as opposed to four calibrated views.

Finally we introduce a new problem of clothing category recognition. We show that the eigen coefficients of clothing deformations are distinctive and can be used to recognize different categories of clothing such as long pants, skirts, short pants, sleeveless tops, etc. Clothing category recognition could be useful for person identification, image search and various retail clothing applications.

In summary, the key contributions of this paper include: 1) the first model of 2D eigen clothing; 2) a full generative 2D model of dressed body shape that is based on an underlying naked model with clothing deformation; 3) the inference of 2D body shape under clothing that uses an explicit model of clothing; 4) shape under clothing in a single image; 5) avoiding “negative clothing” by modeling the skewed statistics of the eigen-clothing coefficients; 6) the first shape-based recognition of clothing categories on dressed humans.

## 2 Related Work

Very little work in computer vision has focused on modeling humans in clothing. What work there is focuses on modeling 3D human shape *under clothing* without actually *modeling* the clothing itself. Bălan and Black [6] present a system based on the 3D SCAPE [8] body model that uses multiple camera views to infer the body shape. They make the assumption that the estimated body shape belongs to a parametric family of 3D shapes that are learned from training bodies. They fit the body to image silhouettes and penalize estimated body shapes that extend beyond the silhouette more heavily than those that are fully inside. This models the assumption that body shape should lie inside the visual hull defined by the clothed body. In essence their method attempts to be robust to clothing by ignoring it. More recently, Hasler *et al.* [7] take a similar approach to fitting a 3D body to laser range scans of dressed humans. Rosenhahn *et al.* [9] model clothing explicitly on a 3D mesh but do so for the purpose of tracking, not body shape estimation. Our approach differs from the above by focusing on 2D models and explicitly modeling clothing deformations on the body using eigen clothing.

The vast majority of work on modeling clothing has focused on the recovery of 3D mesh models of the clothing itself (e.g. [10]). We know of no work on modeling eigen clothing or 2D clothing deformation models. Of course other types of 2D deformable shape models (e.g. active shape models [15]) have been widely used in vision applications and a review is beyond our scope.

Almost all work on 2D person detection and pose estimation implicitly assumes the people are clothed (though [11] is a notable counterexample). Despite this, few authors have looked at using clothing in the process [12] or at actually using a model of the clothing. Recent work by Bourdev and Malik [13] learns body part detectors that include upper and lower clothing regions. They do not model the clothing shape or body shape underneath and do not actually recognize different types of clothing. One recent paper does try to recognize clothing

types for Internet retail applications [14]. The garments, however, are assumed to lie in a plane and are hence not actually on a human body.

### 3 The Contour Person Model

We start with a Contour Person (CP) model [5], which is a low-dimensional, realistic, parameterized generative model of 2D human shape and pose. The CP model is learned from examples created by 2D projections of multiple shapes and poses generated from a 3D body model such as SCAPE [8]. The CP model is based on a template,  $T$ , corresponding to a reference contour that can be deformed into a new pose and shape. This deformation is parameterized and factors the changes of a person’s 2D shape due to pose, body shape, and the parameters of the viewing camera. This factorization allows different causes of the shape change to be modelled separately. Let  $B_T(\Theta) = (x_1, y_1, x_2, y_2, \dots, x_N, y_N)^T$  denote the parametric form of the CP, where  $N$  is the number of contour points and  $\Theta$  is a vector of parameters that controls the deformation with respect to  $T$ . The CP model represents a wide range of 2D body shapes and poses, but only does so for naked bodies. Examples of such body contours,  $B_T(\Theta)$ , are shown in blue in Fig. 1. See Freifeld *et al.* [5] for mathematical details.

The CP model may contain internal or occluded portions of the body contour. However, here our clothing training data consists only of silhouettes with no internal structure. Consequently, we restrict the poses we consider and define  $B_T(\Theta)$  to be a CP model corresponding to a bounding body contour without holes. In future work, we will generalize the DCP model to take advantage of the ability of the CP to accommodate self occlusions.

### 4 Clothing Model

We directly model the deformation from a naked body contour to a clothed body by virtually “dressing” the naked contour with clothing. We start with a training set (described below) of clothing outlines and corresponding naked body outlines underneath. The CP model is first fit to the naked body outline to obtain a CP representation. For each point on the CP, we compute the corresponding point on the clothing outline (described below) and learn a point displacement model using PCA [15]. We further learn a prior over the PCA coefficients using a Beta distribution to prevent infeasible displacements (i.e. “negative clothing”).

The DCP model can be thought of as having two “layers” that decouple the modeling of body pose and shape from the modeling of clothing. The first layer generates a naked body deformation from the template contour and the second layer models clothing deformation from this deformed naked contour. The first layer is the CP model, which is compositional in nature and based on deformations of line segments (see [5]). The second layer, described here, is simpler and is based directly on displacements of contour points. The layered representation is desirable because it allows constraints to be imposed independently on the body and the clothing.

## 4.1 Data Sets

Our method requires training contours of people in clothing for which we know the true underlying naked body shape. We describe two such training sets below.

**Synthetic data set.** Synthetic data provides ground truth body shapes that enable accurate quantitative evaluation. We use 3D body meshes generated from the CAESAR database (SAE International) of laser range scans and dress these bodies in simulated clothing (Fig. 2). We used 60 male and 100 female bodies spanning a variety of heights and weights and use commercial software (OptiTex International, Israel) to generate realistic virtual clothing. The clothing simulation produces a secondary 3D mesh that lies on top of the underlying body mesh by construction. Given a particular camera view, we project the body mesh into the image to extract the body outline and do the same for the combined body and clothing meshes. This provides a pair of training outlines.

For the synthetic dataset we restrict the clothing to a single type (Army Physical Training Uniforms) but in different sizes, as appropriate for the body model. While narrow, this dataset provides nearly perfect training data and ground truth for evaluation.

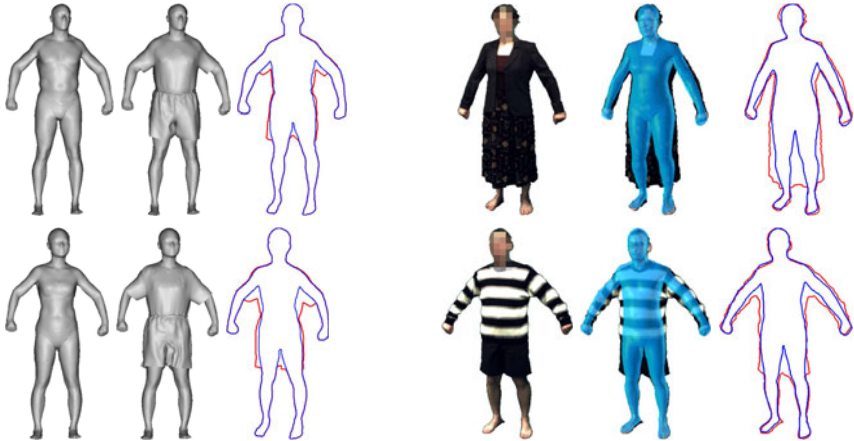
**Real data set.** To model real people in real clothing we use the dataset described by Bălan and Black in [6] (Fig. 2) which contains images of 6 subjects (3 males, 3 females) captured by 4 cameras in two conditions: 1) the “naked condition” in which the subjects wear tight fitting clothing; 2) the “clothed condition” in which they wear different types of “street” clothing. The dataset contains four synchronously captured images of each subject, in each condition, in a fixed set of 11 postures. For each posture the subjects are dressed in 6-10 different sets of clothing (trials). Overall there are 47 trials with a total of 235 unique combinations of people, clothing and poses.

For each image of a dressed person, we use standard background subtraction [6] to estimate the clothed body silhouette and extract the outline. To obtain the underlying naked body contours, we fit a 3D parametric body model using the 4 camera views in the naked condition [6]. We take this estimated 3D body shape to be the true body shape. We then hold this body shape fixed while estimating the 3D pose of the body in every clothing trial using the method of [6] which is robust to clothing and uses 4 camera views.

The process produces a 3D body of the “true” shape, in the correct pose, for every trial. We project the outline of this 3D body into a selected camera view to produce a training 2D body contour. We then pair this with the segmented clothed body in that view. Note that the fitting of the 3D body to the image data is not perfect and, in some cases, the body contour actually lies outside the clothing contour. This does not cause significant problems and this dataset provides a level of realism and variability not found in the synthetic dataset.

## 4.2 Correspondence

Given the naked and clothed outlines defined above, we need to know the correspondence between them. Defining the correspondence between the naked outline



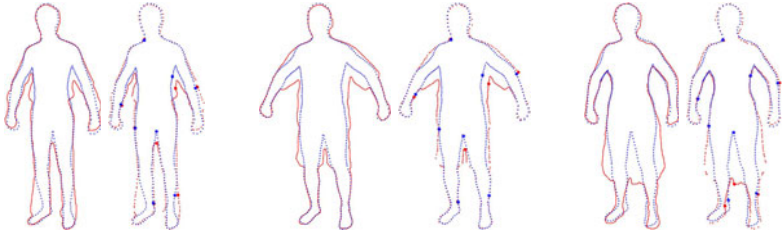
**Fig. 2. Example training data.** Left: Pairs of synthetic 3D bodies, unclad and clothed. Projecting the silhouette contours of these pairs produces training contours. Right: Training contours derived from multi-camera data (see text); the estimated ground truth 3D body is shown as a translucent overlay.

and the clothing outline is nontrivial and how it is done is important. Baumberg and Hogg, for example, model the outline of pedestrians (in clothing) using PCA [17]. In their work, correspondence is simply computed by parameterizing all training contours with a fixed number of evenly sampled points. Incorrect correspondence (i.e. sliding of points along the contour) results in eigen shapes that are not representative of the true deformations of the contours.

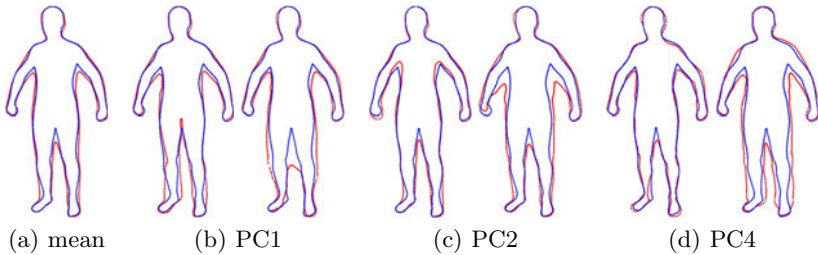
Instead, we start with the trained parametric CP representation  $B_T(\Theta)$  and optimize it to fit the 2D naked body that minimizes the difference between the CP silhouette and the naked body silhouette. This gives a CP representation of the naked body that consists of  $N = 1120$  points. We then densely sample  $M$  points on clothing outline, where  $M \gg N$  and select the  $N$  clothing contour points that best correspond to the CP points. During matching, the relative order of the points is maintained to guarantee the coherence of the match. Let the CP contour be represented by a list of points  $P = \{p_1, p_2, \dots, p_N\}$  and let the sampled clothing outline be represented by  $Q = \{q_1, q_2, \dots, q_M\}$ . We pick a subset of  $N$  points  $G = \{q_{k_1}, q_{k_2}, \dots, q_{k_N}\}$  from  $Q$  that minimizes  $\sum_{i=1}^N \|p_i - q_{k_i}\|^2$  over the indices  $k_i$  such that the ordering,  $k_r < k_s$ , is preserved for  $1 \leq r < s \leq N$ . We use the dynamic programming method proposed in [18]. Example alignments are shown in Fig. 3.

### 4.3 Point Displacement Model

We convert the point list  $G$  to a vector  $\hat{G} = (x_1, y_1, x_2, y_2, \dots, x_N, y_N)^T$  and now we have  $B_T(\Theta)$  for the naked body contour and  $\hat{G}$  for clothing contour, both of which have  $N$  corresponding points. The clothing displacement for a particular training example,  $i$  is then defined by  $\delta_i = \hat{G}_i - B_T(\Theta_i)$ . We collect



**Fig. 3. Correspondence between body and clothing contours.** In each pair: the left image shows the sample points of the body contour in blue and the densely sampled clothing contour in red. The right image shows the final sub-sampled clothing contour with a few matching points highlighted as larger dots. Nearby dots illustrate corresponding points (in some cases they are on top of each other).



**Fig. 4. Eigen clothing.** The blue contour is always the same naked shape. The red contour shows the mean clothing contour (a) and  $\pm 3$  std from the mean for several principal components (b)-(d).

the training displacements into a matrix and perform PCA. We take the first 8 principal components accounting for around 90% of the variance to define the eigen-clothing model. Figure 4 shows the mean and first few clothing eigenvectors for the real data set. This illustrates how the principal components can account for various garments such as long pants, skirts, baggy shirts, etc. Note that simply varying the principal components can produce “negative clothing” that extends inside the blue body contour. We address this in the following section.

Using this model we generate new body shapes in new types of clothing by first sampling CP parameters  $\Theta$  to create a naked body contour  $B_T(\Theta)$  and then using the following equation to generate a clothed body

$$C(\Theta, \eta) = B_T(\Theta) + \Delta_{mean} + \sum_{i=1}^{N_\eta} \eta_i \cdot \Delta_i \quad (1)$$

where  $N_\eta$  is the number of eigenvectors used, the  $\eta_i$ ’s are coefficients,  $\Delta_{mean}$  is the mean clothing displacement, and  $\Delta_i$  is the  $i^{\text{th}}$  eigen-clothing vector.



#### 4.4 Prior on Point Displacement

Although the PCA model captures clothing deformation, it allows point displacements in both inward and outward directions, which violates our assumption that clothing only makes the body appear bigger. This assumption is confirmed by examining the statistics of the linear eigen coefficients in the training data. Figure 5 shows several such distributions, which may be skewed or symmetric. In particular we find that coefficients for the principal components that capture the most variance are typically positively or negatively skewed while coefficients for the lower-variance components tend to be more normally distributed. The first few eigenvectors capture the gross clothing displacements, which are always away from the body. Of course clothing also exhibits many fine details and folds and these are captured by the lower variance eigenvectors. These “detail” eigenvectors modify the main clothing contour both positively and negatively (e.g. out and in) and hence tend to have more symmetric statistics.

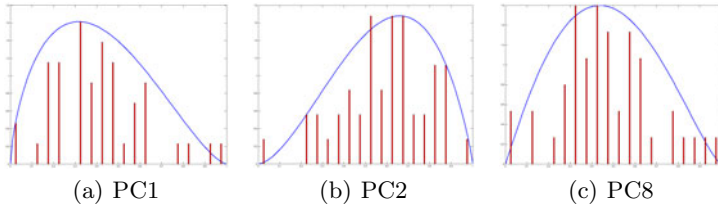
Based on the observation of natural clothing statistics, we learn a prior on the PCA coefficients to penalize infeasible clothing displacements. We make the assumption that the eigenvectors are independent (not necessarily true since the data is not Gaussian) and independently model a prior on each coefficient using a Beta distribution. The Beta distribution is defined on  $[0, 1]$  and is characterized by two parameters  $\alpha$  and  $\beta$  that can be varied to capture a range of distributions including positively skewed, negatively skewed and symmetric shapes:

$$\mathbf{Beta}(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}. \quad (2)$$

Given  $L$  training body/clothing pairs, and the associated clothing displacements, we project each displacement onto the PCA space to obtain coefficient  $\eta_m^l$  for instance  $l$ , ( $l \in [1, L]$ ), on eigenvector  $m$ . We normalize  $\eta_m^1, \eta_m^2, \dots, \eta_m^L$  to  $[0, 1]$  to obtain  $\tilde{\eta}_m^1, \tilde{\eta}_m^2, \dots, \tilde{\eta}_m^L$  and fit these with the Beta distribution. The probability of observing a normalized coefficient  $\tilde{x}_m$  for the  $m^{\text{th}}$  eigenvector is given by  $\mathbf{Beta}(\tilde{x}_m, \alpha_m, \beta_m)$ , where  $\alpha_m$  and  $\beta_m$  are the estimated parameters of the Beta distribution. If we observe a coefficient during testing that is out of the scope of the training coefficients, we threshold it to be between the minimal and maximal value in the training set and normalize it to compute its prior probability. If thresholded, however, we still use the original value to reconstruct the shape. Figure 5 shows how the Beta function can represent a variety of differently shaped distributions of clothing displacement coefficients.

#### 4.5 Inference

The inference problem is to estimate the latent variables  $\Theta$  and  $\eta$  by only observing a single image of a person in clothing. We define a likelihood function in terms of silhouette overlap. We adopt a generative approach in which  $C(\Theta, \eta)$ , the clothed body (Eq. 1), defines an estimated silhouette,  $S^e(C(\Theta, \eta))$ , and compare it with the observed image silhouette  $S^o$ . We follow [6] and define the asymmetric distance between silhouettes  $S^r$  and  $S^t$  as  $d(S^r, S^t) = \frac{\sum_{i,j} S_{i,j}^r H_{i,j}(S^t)}{\sum S_{i,j}^r}$ , where



**Fig. 5. The statistics of clothing displacements.** Example histograms and Beta distribution fits to linear eigen-clothing coefficients. Note the skew that results from the fact that clothing generally makes the body appear larger.

$S_{i,j}^r$  is a pixel inside silhouette  $S^r$  and  $H_{i,j}(S^t)$  is a distance function which is zero if pixel  $(i, j)$  is inside  $S^t$  and is the distance to the closest point on the boundary of  $S^t$  if it is outside.

We then define the data term as the following symmetric data error function

$$E_{data}(\Theta, \eta) = d(S^e(C(\Theta, \eta)), S^o) + d(S^o, S^e(C(\Theta, \eta))). \quad (3)$$

The first part of Eq. 3 penalizes the regions of the synthesized clothing instance  $S^e(C(\Theta, \eta))$  that fall outside the observed clothing silhouette  $S^o$ , and the second part makes  $S^e(C(\Theta, \eta))$  explain  $S^o$  as much as possible.

$E_{data}$  alone is not sufficient to estimate  $\Theta$  and  $\eta$  correctly, because there are ambiguities in estimating smaller bodies with larger clothing and larger bodies with smaller clothing. As was mentioned in Sec. 4.4, we use the Beta prior to penalize unlikely displacements. Recall that  $\tilde{\eta}_m$  represents the normalized coefficient for the  $m^{\text{th}}$  basis. The prior term is defined as

$$E_{prior}(\eta) = - \sum_m \log(\mathbf{Beta}(\tilde{\eta}_m, \alpha_m, \beta_m)). \quad (4)$$

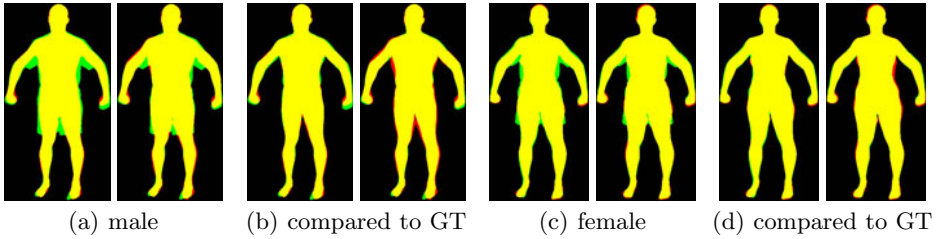
The final energy function we minimize is

$$E(\Theta, \eta) = E_{data}(\Theta, \eta) + \lambda E_{prior}(\eta) \quad (5)$$

where  $\lambda$  indicates the importance of the prior. Problems with “negative clothing” and clothing that is unusually large are avoided due to the prior. Optimization is performed using MATLAB’s `fminsearch` function.

## 5 Results

We consider two novel applications of the proposed method. The first is the estimation of 2D body shape under clothing given a single image of a clothed person. The second is the recognition of different clothing categories by classifying the estimated clothing deformation parameters. We evaluate our model on three tasks: body shape estimation from synthetic data, body shape estimation from real data, and clothing type classification from real data. We compare



**Fig. 6. Synthetic data results.** For each pair of images, the DCP result is on the left and NM result is on the right. The first pair shows an estimated body silhouette (red) overlaid on the clothing silhouette (green); overlapped regions are yellow. The second pair shows the estimated body (red) overlaid on the ground truth (GT) body (green). The third and fourth pairs show the same but for a female. NM typically overestimates the size of the body.

the results of the first two tasks with approaches that do not explicitly model clothing deformation.

**Body estimation under clothing from synthetic data.** We use the synthetic dataset of 60 males and 100 females, in and out of synthetic clothing, as described above. We randomly select 30 males and 50 females as the training set and the remaining 80 bodies as the test set. A gender-specific CP model is learned for males and females separately while a gender-neutral eigen model is learned for clothing deformations. We estimate the underlying bodies for the test samples using the *Dressed Contour Person* (DCP) and measure the estimation error as

$$\text{err}(S^{EST}, S^{GT}) = \frac{\sum_{i,j} |S_{i,j}^{EST} - S_{i,j}^{GT}|}{2 \sum_{i,j} S_{i,j}^{GT}} \quad (6)$$

where  $S^{EST}$  is a silhouette corresponding to the estimated naked body contour and  $S^{GT}$  is the ground truth underlying naked body silhouette. The results of DCP are also compared with a naive method (NM) in which we simply fit the CP model to the image observations of clothed people. As in [6], the NM attempts to account for clothing by penalizing contours more if the estimated body silhouette falls outside of the clothing observation than if it does not fully explain the clothing observation. The average estimation errors obtained with NM for males and females are 0.0456 and 0.0472 respectively while DCP achieves 0.0316 and 0.0308. Our DCP model improves accuracies over NM by 30% (male) and 35% (female) relatively. While the synthetic dataset has only one clothing type, the bodies span a wide range of shapes. The results show a principled advantage to modeling clothing deformation compared with ignoring clothing. Figure 6 shows some representative results from the test set.

**Body estimation under clothing from real data.** Figure 7 shows 8 different poses from the real dataset (Sec. 4.1). For each pose there are 47 examples having unique combinations of subjects and clothing types. Since the number of body/clothing pairs is limited in each pose, we use a leave-one-out strategy where

**Table 1.** Comparison on real data: DCP, NM, and NP3D methods (see text)

Method, AAE	Pose1	Pose2	Pose3	Pose4	Pose5	Pose6	Pose7	Pose8	Average
DCP	0.0372	0.0525	0.0508	0.0437	0.0433	0.0451	0.0503	0.0668	0.0487
NP3D	0.0411	0.0628	0.0562	0.0484	0.0494	0.046	0.0472	0.0723	0.0529
NM	0.0865	0.0912	0.0846	0.0835	0.0877	0.0921	0.0902	0.1184	0.0918
Significance ( $p$ -value)									
DCP vs NP3D	0.38	0.13	0.34	0.46	0.36	0.89	0.66	0.54	0.07
DCP vs NM	6.4e-7	4.9e-4	2.1e-4	2.1e-4	6.7e-8	1.0e-5	1.0e-6	2.3e-4	9.9e-17

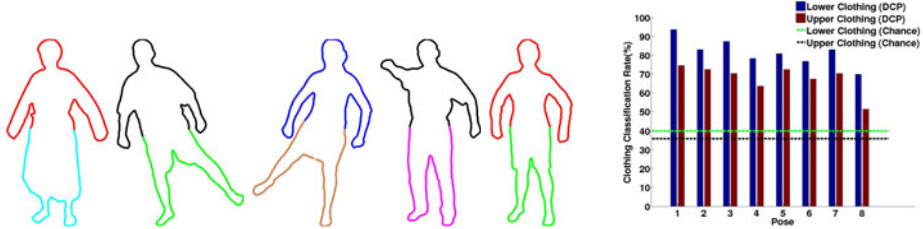


**Fig. 7. Sample DCP results of estimated body shape overlaid on clothing.** The estimated body contour and synthesized clothing contour are depicted by blue and red outlines respectively. Body shape is the transparent region encompassed by the body contour. Results are shown for a variety of poses (left to right: 1-8) and viewing directions.

we estimate the body of instance  $i$  using the eigen-clothing model learned from all remaining 46 instances excluding  $i$ . We use DCP to estimate the underlying body shape for a total of  $47 * 8 = 376$  instances (Fig. 7) and compare the results with two other methods: 1) NM described in the previous experiment; and 2) “Naked People estimation in 3D” (NP3D) proposed in [6]. Since DCP and NM are 2D methods using a 2D CP model, they only use one camera view. NP3D, however, estimates a 3D body model from four camera views [6]. To compare with NP3D we project the estimated body from NP3D into the image corresponding to the camera view used by our method.



**Fig. 8. Comparisons of DCP, NM, and NP3D.** For each group of images: the first 3 images (left to right) show overlap of the estimated silhouette (red) and the ground truth silhouette (green) for DCP, NP3D, and NM (yellow is overlap); the 4th image shows the body estimated by NM overlaid on a clothing image. NM overestimates body shape as expected.



**Fig. 9. Color coded clothing type.** We consider three types of upper clothing: long sleeves (red), short sleeves (black) and sleeveless tops (blue) and four types of lower clothing: short pants (green), long pants (magenta), short skirts (coffee), and long skirts (cyan). Classification results for the 7 clothing types in all 8 poses are shown in the right figure compared to “Chance”.

Table 1 shows the Average Estimation Error (AEE) computed by averaging  $err(\cdot, \cdot)$  (Eq. 6) over the 47 instances for each pose (or over all poses in the last column). Figure 8 shows details of the fitting results. We find that DCP has lower error than both NM and NP3D. In the case of NM these differences are statistically significant (paired t-test,  $p < 0.05$ ) for all poses and in the aggregate. While DCP has lower error than NP3D in all but one pose, and lower error overall, the differences are not significant at the  $p < 0.05$  level. Recall that NP3D is using significantly more information. These results suggest that using a learned statistical model of clothing is preferable to simply trying to ignore clothing [6].

**Clothing category recognition.** We now ask whether the clothing deformation coefficients contain enough information about clothing shape to allow the classification of different types of clothing. Note that this task involves recognizing clothing *on the body* as it is worn by real people. We separate upper clothing and lower clothing and define 7 different categories (as color coded in Fig. 9).

We use a simple nearest neighbor (NN) classifier with Euclidean distances computed from the coefficients along the first 8 principal components. Since we have a limited number of clothing instances (47) for each pose, we use a

leave-one-out strategy and assume that we know the categories of all the instances except the one we are testing. Each instance is then assigned a category for both upper clothing and lower clothing based on its nearest neighbor. Classification results are shown in Fig. 9 along with chance performance for this task.

## 6 Conclusions

We have presented a new generative model of the 2D human body that combines an underlying Contour Person representation of the naked body and layers on top of this a clothing deformation model. This goes beyond previous work to learn an eigen model of clothing deformation from examples and defines a prior over possible deformations to prevent “negative clothing”. While previous work has examined 3D body models captured with multiple cameras or laser range scanners, we argue that many computer vision applications use 2D body models and that these applications will benefit from a more realistic generative model of clothed body shape. By modeling clothing deformations we estimate 2D body shape more accurately and even out-perform previous multi-camera systems on estimating shape under clothing. Finally we define a new problem of clothing category recognition on the human body and show how the coefficients of the estimated eigen clothing can be used for this purpose. This new dressed person model is low dimensional and expressive, making it applicable to many problems including 2D human pose estimation, tracking, detection and segmentation.

Our method does have some limitations. The method assumes there is a correspondence between body contour points and clothing contour points. When there is significant limb self occlusion, the clothing silhouette may not contain features that *correspond to* that limb. Dealing with significant self occlusion is future work. Also, here we assume that the rough viewing direction (frontal or side) and rough pose are known.

There are several directions for future work. First, we plan to model clothing deformation as a function of human movement. This may require a model more like the original CP model in which deformations are defined as scaled rotations of contour line segments [5]. This representation allows the factoring of contour changes into different deformations that can be composed. Second, we will explore what we call “eigen separates”; that is, separate eigen models for tops and bottoms as well as for hair/hats and shoes. Having separate eigen spaces reduces the amount of training data required to capture a wide range of variations. Finally we plan to extend these methods to model 3D clothing deformations from a 3D body model. Again data acquisition for 3D clothed and unclothed training data is very difficult, and we plan to use realistic physics simulation of clothing.

**Acknowledgements.** This work was supported in part by NIH EUREKA 1R01NS066311-01 and NSF IIS-0812364. We thank L. Reiss for generating the synthetic training data, A. Bălan for assistance with the real dataset, and S. Zuffi for helpful comments.

## References

1. Mori, G., Ren, X., Efros, A., Malik, J.: Finding and tracking people from the bottom up. In: CVPR, vol. 2, pp. 467–474 (2003)
2. Felzenszwalb, P.F.: Representation and detection of deformable shapes. PAMI 27, 208–220 (2004)
3. Hinton, G.E.: Using relaxation to find a puppet. In: Proc. of the A.I.S.B. Summer Conference, pp. 148–157 (1976)
4. Ju, S.X., Black, M.J., Yacoob, Y.: Cardboard people: A parameterized model of articulated motion. *Face and Gesture*, 38–44 (1996)
5. Freifeld, O., Weiss, A., Zuffi, S., Black, M.J.: Contour people: A parameterized model of 2D articulated human shape. In: CVPR (2010)
6. Balan, A., Black, M.: The naked truth: Estimating body shape under clothing. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 15–29. Springer, Heidelberg (2008)
7. Hasler, N., Stoll, C., Rosenhahn, B., Thormählen, T., Seidel, H.: Estimating body shape of dressed humans. *Computers & Graphics* 33, 211–216 (2009)
8. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: shape completion and animation of people. *ACM TOG (SIGGRAPH)* 24, 408–416 (2005)
9. Rosenhahn, B., Kersting, U., Powell, K., Klette, R., Klette, G., Seidel, H.: A system for articulated tracking incorporating a clothing model. *Mach. Vis. App.* 18, 25–40 (2007)
10. White, R., Crane, K., Forsyth, D.: Capturing and animating occluded cloth. *ACM TOG (SIGGRAPH)* 26, 34 (2007)
11. Fleck, M., Forsyth, D., Bregler, C.: Finding naked people. In: ECCV, vol. 2, pp. 592–602 (1996)
12. Sprague, N., Luo, J.: Clothed people detection in still images. In: ICPR, vol. 3, pp. 585–589 (2002)
13. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3D human pose annotations. In: ICCV (2009)
14. Tseng, C.H., Hung, S.S., Tsay, J.J., Tsaih, D.: An efficient garment visual search based on shape context. *W. Trans. on Comp. (WSEAS)* 8, 1195–1204 (2009)
15. Cootes, T., Taylor, C.: Active shape models—smart snakes. In: BMVC, pp. 266–275 (1992)
16. Balan, A., Sigal, L., Black, M., Davis, J., Haussecker, H.: Detailed human shape and pose from images. In: CVPR, pp. 1–8 (2007)
17. Baumberg, A., Hogg, D.: Learning flexible models from image sequences. In: ECCV, pp. 299–308 (1994)
18. Oliveira, F., Tavares, J.: Algorithm of dynamic programming for optimization of the global matching between two contours defined by ordered points. *Computer Modeling in Eng. & Sciences* 31, 1–11 (2008)
19. Sigal, L., Balan, A., Black, M.: Combined discriminative and generative articulated pose and non-rigid shape estimation. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 20, pp. 1337–1344. MIT Press, Cambridge (2008)

# Self-Adapting Feature Layers

Pia Breuer and Volker Blanz

Institute for Vision and Graphics – University of Siegen  
{pbreuer,blanz}@informatik.uni-siegen.de

**Abstract.** This paper presents a new approach for fitting a 3D morphable model to images of faces, using self-adapting feature layers (SAFL). The algorithm integrates feature detection into an iterative analysis-by-synthesis framework, combining the robustness of feature search with the flexibility of model fitting. Templates for facial features are created and updated while the fitting algorithm converges, so the templates adapt to the pose, illumination, shape and texture of the individual face. Unlike most existing feature-based methods, the algorithm does not search for the image locations with maximum response, which may be prone to errors, but forms a tradeoff between feature likeness, global feature configuration and image reconstruction error.

The benefit of the proposed method is an increased robustness of model fitting with respect to errors in the initial feature point positions. Such residual errors are a problem when feature detection and model fitting are combined to form a fully automated face reconstruction or recognition system. We analyze the robustness in a face recognition scenario on images from two databases: FRGC and FERET.

## 1 Introduction

Fitting generative models such as 3D morphable models (3DMM) or active appearance models (AAM) to images of faces has turned out to be a promising approach to obtain a face-specific encoding of faces for recognition purposes. Due to the 3D representation, 3DMMs can help to recognize faces at arbitrary poses and illuminations [1]. A bottleneck in the development of automated fitting algorithms is the initialization of the optimization. While early work has started from a coarse alignment [2], later versions have relied on manually defined feature point positions [1]. Recently, a fully automated 3DMM fitting algorithm has been presented [3] which uses Support Vector Machines (SVM) for the detection of faces and facial features. However, the quality of the fit turned out to depend critically on the precision of the facial features. The goal of this paper is to integrate feature detection into the 3DMM fitting procedure.

In order to leverage the fact that 3DMM fitting can be applied to any pose and illumination, it is important to have feature detectors that are either invariant, or to rely on a set of different detectors, or - as we propose here - to have adaptive feature detectors. In our approach, the feature detector is updated by rendering an image of the current estimate of the face at the current estimate of the imaging parameters several times during the optimization, and forming



templates from predefined face regions. Unlike more powerful feature detectors such as SVM or AdaBoost [4], template matching (TM) does not require multiple training samples.

The second contribution of this paper is a novel way to include facial features into model fitting. Most existing algorithms find the image position with maximum response of the feature detector and pull the corresponding point of the model towards this position. However, on difficult images, it may occur that the feature response at the correct position is not the global maximum. Therefore, we propose a strategy that forms a tradeoff between high feature detector response and a plausible overall configuration. This is achieved by including the value of the feature detector response as an additional term in the cost function of 3DMM fitting, rather than the 2D distance between the current feature position of the model and the position of the global maximum. Each feature detector response forms an additional 2D array, or layer, that is used along with the three color channel layers which form the image. On a more general level, the approach introduces a new, high-level criterion for image similarity to analysis-by-synthesis strategies. In fact, this can be implemented with any feature detector or any other local descriptor of image properties.

## 2 Related Work

Detection of facial features and integration into face recognition systems have been studied extensively in recent years. Still, robust feature detection in difficult imaging conditions continues to be a challenge.

AdaBoost [4] is a well-known approach for face and facial feature detection. [5] use it to first detect candidates for eyes, nose and lips separately. From the candidates, the combination with highest log-likelihood is chosen. Many approaches use coarse-to-fine strategies: [6] detect the head using AdaBoost and get a first guess of the iris position using linear regression. At the next step a weighted support vector machine (SVM), using only a small number of pixels of the whole search area, refines the iris position. [7] use a cascade of global deformation, texture fitting and feature refinement to refine eye, mouth, nose and eyebrow positions. [8] use a hierarchical face model composed of a coarse, global AAM and local, detailed AAMs for each feature for refinement. This restricts the influence of noise to the features directly nearby, and prevents it from affecting the rest of the face. [9] and [10] both use a prior distribution map analyzing AdaBoost face detection output as a starting condition, and refine the feature positions using color values and a decision tree classifier [9], or using a HarrisCornerDetector and a SVM to classify whether the detected corners belong to a feature or not [10]. [11] find facial features indirectly by using templates for parts of the face in connection with vectors that point from these regions to the positions of the features. The final feature positions are weighted combinations of the vectors.

Instead of refining the positions as in a coarse-to-fine approach, [12] combine conventional algorithms in a sequence to get better initial values for characteristic points: face detection by skin-color and luminance constraints, eye detection

by TM and symmetry enforcement, mouth and eyebrow detection both using luminance and geometry constraints. Other combined approaches have been proposed by [13] who classify SURF local descriptors with SVMs: one SVM to decide whether they belong to the face or not, followed by special SVMs for each feature. [14] combine four feature detectors (DCT, GaborWavelets, ICA, non-negative Matrix Factorization) on images at a reduced resolution. SVM is performed to get the most reliable positions (highest SVM scores) for each feature, and a graph based post-processing method is used to pick the best combination of feature positions. Refinement of the feature positions at the end is done using DCT again on full resolution.

A number of algorithms introduce local features to Active Shape Models (ASM) and Active Appearance Models (AAM): [15] extend the ASM by fitting more landmarks, using 2D-templates at some landmark positions and relaxing the shape model wherever it is advantageous. To improve the result further, they use the first alignment as start value for a second fitting with the new ASM. [16] combine ASM and Haar wavelets. [17] use a similar approach to ours, yet their 2D AAM model is designed for frontal or nearly frontal views of faces only. They form facial feature detectors from an AAM and update them in an iterative search algorithm. In each iteration, they find the feature positions with a plausible 2D configuration (high prior probability) and, at the same time, a high feature detector output. In contrast, we use a 3D model that contains additional parameters for pose and illumination, use different methods to create feature detectors and to fit the model, and we integrate the feature point criterion into a cost function that includes overall image difference for a global analysis-by-synthesis.

The first combination of feature detection with 3D morphable models (3DMM) was presented by [18] who created local feature detectors for face recognition from a 3DMM. Unlike our approach, they first reconstructed a face from a gallery image, created virtual images using the 3DMM and then relied on SVM-based local classifiers for recognition. [19] presented a patch based approach that is related to ours because it combines local feature detectors and a 3D shape model. In contrast to our algorithm, however, the feature detectors are trained prior to fitting, and the model fitting minimizes the 2D distances between image points with maximum response of the feature detectors and the corresponding model points. [20] first identify all potential feature points in the image by using SIFT as a criterion for saliency, then reject those that are similar to none of the points in the appearance model, and subsequently find the configuration of features in the image and the mapping to features of the 3DMM that has a maximum likelihood. The resulting feature locations can be used to initialize a 3DMM fitting procedure. [3] use SVM for detecting faces, estimating pose angles and finding facial features. From a number of nose point candidates, a model-based criterion selects the most plausible position. Then, these data are used to initialize a 3DMM fitting algorithm and compute 3D reconstructions. Our approach may be used in a similar general approach, but with an increased robustness to unprecise initial feature positions. In a Multi-Features Fitting Algorithm for the 3DMM, [21] use a cost function that adds color difference, edge information and

the presence of specular highlights in each pixel. The algorithm is related to ours because multiple features are used and a tradeoff is found for the match of each feature plus a prior probability term. However, we use features that are derived from facial appearance, and we update these features during the optimization.

### 3 Morphable Model of 3D Faces

For the reconstruction of a high-resolution 3D mesh, we use a Morphable Model of 3D faces (3DMM, [2]), which was built by establishing dense correspondence on scans of 200 individuals who are not in the test sets used below. Shape vectors are formed by the  $x, y, z$ -coordinates of all vertices  $k \in \{1, \dots, n\}$ ,  $n = 75,972$  of a polygon mesh, and texture vectors are formed by red, green, and blue values:

$$\mathbf{S} = (x_1, y_1, z_1, x_2, \dots, x_n, y_n, z_n)^T \quad (1)$$

$$\mathbf{T} = (R_1, G_1, B_1, R_2, \dots, R_n, G_n, B_n)^T. \quad (2)$$

By Principal Component Analysis (PCA), we obtain a set of  $m$  orthogonal principal components  $\mathbf{s}_j$ ,  $\mathbf{t}_j$ , and the standard deviations  $\sigma_{S,j}$  and  $\sigma_{T,j}$  around the averages  $\bar{\mathbf{s}}$  and  $\bar{\mathbf{t}}$ . In this paper, only the first 99 principal components of shape and texture are used, because they cover most of the variance observed in the training set. A larger number would increase the computation time while not improving the results significantly.

In an analysis-by-synthesis loop, we find the face vector from the Morphable Model that fits the image best in terms of pixel-by-pixel color difference between the synthetic image  $I_{model}$  (rendered by standard computer graphics techniques), and the input image  $I$ :

$$E_I = \sum_{x,y} (I(x,y) - I_{model}(x,y))^2. \quad (3)$$

The squared differences in all three color channels are added in  $E_I$ . We suppress the indices for the separate color channels throughout this paper. The optimization is achieved by an algorithm that was presented in [12]. In each iteration, the algorithm evaluates  $E_I$  not on the entire image, but only on 40 random vertices. For the optimization to converge, the algorithm has to be initialized with the feature coordinates of at least 5 feature points.

The goal is to minimize the cost function

$$\mathbf{E} = \eta_I \cdot E_I + \eta_M \cdot E_M + \eta_P \cdot E_P \quad (4)$$

where  $E_M$  is the sum of the squared distances between the 2D positions of the marked feature points in the input image, and their current positions in the model.  $E_P$  is the Mahalanobis distance of the current solution from the average face, which is related to the log of the prior probability of the current solution.  $\eta_I$ ,  $\eta_M$  and  $\eta_P$  are weights that are set heuristically: The optimization starts with a conservative fit ( $\eta_M$  and  $\eta_P$  are high), and in the final iterations  $\eta_M = 0$  and  $\eta_P$  is small.

The algorithm optimizes the linear coefficients for shape and texture, but also 3D orientation and position, focal length of the camera, angle, color and intensity

of directed light, intensity and color of ambient light, color contrast as well as gains and offsets in each color channel.

## 4 Self-Adapting Features

Our proposed self-adapting feature approach is built on top of the 3DMM and introduces a novel criterion in the cost function. The goal is to reduce the influence of the (potentially unreliable) initial feature positions that are used in  $E_M$ : they are only used for the first coarse alignment of the head, and discarded later. After coarse alignment and a first estimation of the illumination, the term  $E_M$  in the cost function (Eqn. 4) is replaced by new  $E_{F_i}$ , that will be explained below, with  $i = 1 \dots 7$ , for the set of 7 feature positions to be refined, weighted with  $\eta_F$ :

$$\mathbf{E} = \eta_I \cdot E_I + \eta_F \cdot \sum_{i=1}^7 E_{F_i}(x_{F_i}, y_{F_i}) + \eta_P \cdot E_P \quad (5)$$

The features are: the tip of the nose, the corners of the mouth, and the inner and outer corners of the eyes. For feature point  $i$ , we know which vertex  $k_i$  of the model it corresponds to, and using perspective projection we get the current position  $(x_{F_i}, y_{F_i})$  in the image  $I_{model}$ .

Once every 1000 iterations, the entire current fitting result  $I_{model}$  is rendered, and templates are cut out around the current feature positions  $(x_{F_i}, y_{F_i})$ . Template sizes are pre-defined relative to the head size  $s_H$  (distance between a vertex on the top of the forehead and one on the bottom of the chin, in pixel units): eyes:  $(\frac{1}{9}s_H) \times (\frac{2}{9}s_H)$ , nose:  $(\frac{1}{18}s_H) \times (\frac{1}{18}s_H)$  and mouth:  $(\frac{2}{9}s_H) \times (\frac{1}{9}s_H)$ . We chose these sizes to make sure that each template contained enough diagnostic features, such as part of the eyebrows in the eye template.

The new  $E_{F_i}$  in (5), based on TM, are

$$\mathbf{E}_{F_i}(x_{F_i}, y_{F_i}) = 1 - \mathbf{C}_{F_i}(x_{F_i}, y_{F_i}). \quad (6)$$

where  $C_{F_i}$  is the normalized cross correlation [22], which we found to be more reliable than alternative choices:

$$\mathbf{C}_{F_i}(x, y) = \frac{\sum_{(p,q) \in R} (I(x+p, y+q) \cdot R(p, q)) - N \cdot \bar{I}(x, y) \cdot \bar{R}}{\sqrt{\sum_{(p,q) \in R} (I(x+p, y+q))^2 - N \cdot (\bar{I}(x, y))^2 \cdot \sigma_R}} \quad (7)$$

where  $I$  is the original image and  $\bar{I}(x, y)$  its local mean value around the current position  $(x, y)$  in a template-sized area,  $R$  is the current template (or reference image) and  $\bar{R}$  its mean value (over all  $(p, q)$ ),  $\sigma_R$  is the variance of the template values and  $N$  is the number of template values ( $width \cdot height$ ). Only  $\bar{I}$  has to be computed for every  $(x, y)$ . The other three components ( $\bar{R}$ ,  $\sigma_R$ ,  $N$ ) can be precomputed. Note that  $\forall (x, y) \in I : \mathbf{C}_{F_i}(x, y) \in [-1, 1]$ , with 1 representing a maximum match and  $-1$  a maximum mismatch. For color images,  $\mathbf{C}_{F_i}(x, y) = \frac{1}{3}(\mathbf{C}_{F_i,red}(x, y) + \mathbf{C}_{F_i,green}(x, y) + \mathbf{C}_{F_i,blue}(x, y))$ .

The weight  $\eta_F$  is constant and scales the sum of all  $E_{F_i}$  to the same range of values as the image difference  $E_I$ .

Templates  $R$  and cross correlations  $\mathbf{C}_{\mathbf{F}_i}$  are updated once every 1000 iterations of the fitting algorithm. To reduce computation time,  $\mathbf{C}_{\mathbf{F}_i}(x, y)$  for each feature  $i$  is calculated only in a region of interest (ROI:  $\frac{1}{9}s_H \times \frac{1}{9}s_H$ ) around the current position  $(x_{F_i}, y_{F_i})$ . Even in the first iteration, these positions can be assumed to be approximately correct, and also the head size  $s_H$  that defines the (constant) template size will be in the right order of magnitude, due to the vague initial feature coordinates.

In intermediate iterations,  $\mathbf{C}_{\mathbf{F}_i}$  remain fixed, but the positions  $(x_{F_i}, y_{F_i})$  for looking up  $\mathbf{C}_{\mathbf{F}_i}(x_{F_i}, y_{F_i})$  will change. This reflects the fact that the locations of feature points may change faster during the optimization than the appearances of features do. Fig. 1 gives an overview of the algorithm.

Fig. 2 shows how the templates (here: outer corner of the left eye) change over fitting iterations when fitting to different images (rows in Fig. 2). Over the first six template-adaptions, not much change is observed. At the seventh template in each row, the change is already visible at the eyebrow, after the eighth and ninth adaption the whole templates changed significantly. The major change can be observed at step eight and nine, because this is where fine adjustment starts: The head model is broken down in different regions, and these are optimized separately (see 2).

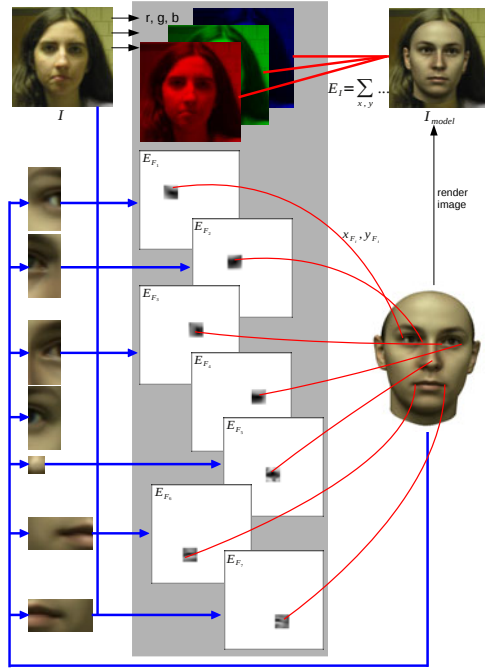
Fig. 3 shows an example of how the cross correlation result, matching the left corner of the left eye, changes over the fitting iterations. The detail belongs to the result of the third line of Fig. 8. There, first the position of the corner of the eye has been displaced to the right to evaluate robustness. As the fitting proceeds, it moves to the left and upward until it reaches the correct position eventually. The position of the ROI also shows where the feature has been positioned when computing the cross correlation. The ROI position reveals the drift of the feature to the correct position.

It can be seen that the cross correlation turns into a single, wide optimum as the template adapts to the appearance in the image. Note that if the model adapts perfectly to the feature in the input image,  $\mathbf{C}_{\mathbf{F}_i}(x, y)$  will converge to the autocorrelation function, and the width of maxima and minima will be determined by the frequency spectrum of the template.

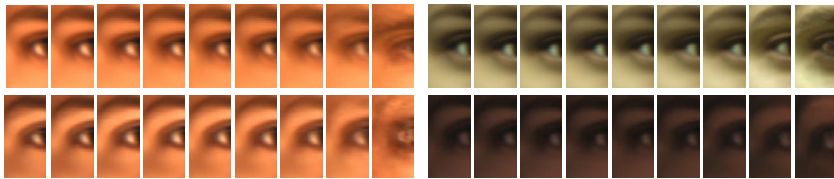
## 5 Results

We tested our algorithm on 300 randomly chosen images from the FRGC data base [23], using three images per person and a set of 50 women and 50 men. The only constraint in random selection was that the person did not show an extreme facial expression. Typical examples of the randomly chosen samples, some of them in difficult imaging conditions (focus, illumination, expressions) can be found in Fig. 4. The database contains front view images only. We show results on non-frontal views later in this section.

For ground truth in every image, five feature positions (outer corners of the eyes, nose and corners of the mouth) were labelled manually. To simulate scenarios with an unreliable initial feature detector, we perturbed the feature positions randomly:



**Fig. 1. Self-adapting feature layers:** Blue arrows show actions performed only every 1000 iterations: Templates are cut out from the current fitting result. They are compared to the original image  $I$  using normalized cross correlation at a certain ROI and from these, the 'feature layers' are generated. Red lines show actions performed every iteration:  $I_{model}$  is compared to  $I$ , and for each feature  $i$  the error value  $E_{F_i}$  is taken from the corresponding 'feature layer' at the current feature position  $(x_{F_i}, y_{F_i})$ .



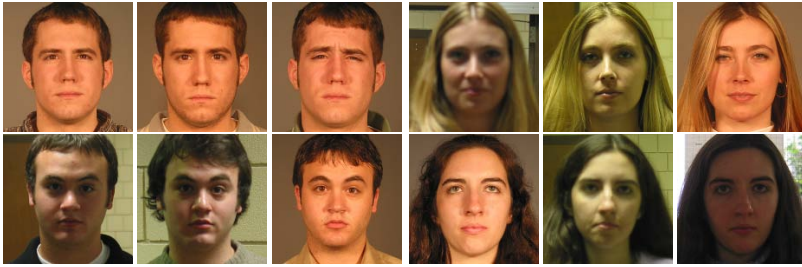
**Fig. 2. Templates** changing over fitting iterations. Each line is from fitting to one input image. In the first two examples (upper row), convergence was correct, in the last two (lower row), the corner of the eye moved to the eyebrow.

1. randomly select two (of the five) features to be perturbed
2. randomly select a displacement direction for each
3. displace feature positions by a fixed distance

In three different test conditions, we used distances of 5%, 12% and 25% of the vertical distance between eyes and nose. This corresponds to distances of 0.2cm, 0.48cm and 1.0cm in reality on an average sized head. The perturbation ranges



**Fig. 3. Cross correlation results in the ROI, changing over fitting iterations.** Dark pixels indicate good matches. These results correspond to the templates in the first line of Fig. 2).



**Fig. 4. Typical examples of images per person:** Each line shows three pictures of the same individual [23]. Here not the whole pictures, but only the facial regions (scaled to the same size) are shown.

are visualized as the radii of the circles in the upper row of Fig. 5. The lower row shows a typical example for each test condition. By using displacement distances relative to the eye-nose distance in the image, rather than fixed pixel distances, we were able to use images at different resolutions.

To have an independent criterion for the quality of the reconstructions, we evaluate recognition rates from model coefficients in an identification task. Given the linear 3DMM coefficients for shape and texture of the entire face and the facial regions (eyes, nose, mouth and surrounding area), which are concatenated into coefficient vectors  $\mathbf{c}$ , the algorithm finds the individual from the gallery set with a minimum distance, measured in terms of a cosine criterion  $d = \frac{\langle \mathbf{c}_1, \mathbf{c}_2 \rangle}{\|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\|}$  (see [13]). For each probe image, a comparison with the other two images of that person and with all three images of all 99 other individuals is performed.

Recognition is tested with the standard 3DMM fitting algorithm ([1], see Section 3) and with our new SAFL approach for the manually marked feature positions and each perturbation range. The percentages of correct identification can be found on the left side of Fig. 7.

Due to the difficult imaging conditions, the overall recognition rate is below 50%. In the unperturbed case, both the standard algorithm and the new self-adapting feature layers (SAFL) deliver similar results, indicating that SAFL do not downgrade the system when correct feature positions are given. However, with perturbed features, the recognition rate for the standard algorithm rapidly decreases as the displacements get larger. In contrast, SAFL identification rates remain stable. This demonstrates that SAFL increases the robustness of the fitting for face recognition.



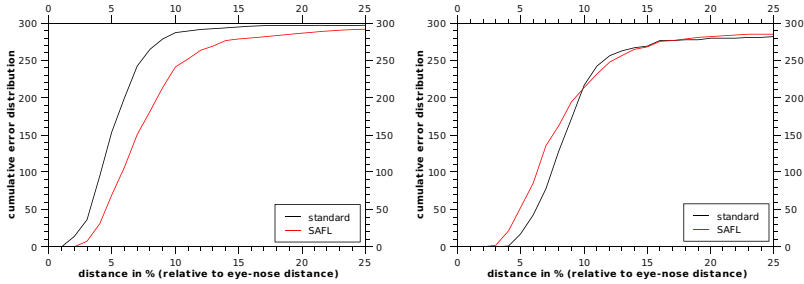
**Fig. 5. Perturbation ranges and typical examples of perturbed positions:** In the upper row circles mark the perturbation ranges of the three different test conditions, from left to right: 5%, 12% and 25%. In the lower row green crosses mark manually labelled feature positions and red crosses mark perturbed positions.

We have also evaluated the distances between the ground truth feature positions and the optimized positions after fitting. Fig. 6 shows the distribution of the average 2D distances of the five features in each test image: The vertical axis is the absolute number of test images (out of a total of 300) where the average feature distance is below the distance threshold indicated on the horizontal axis.

If we do not perturb the starting positions of features, most test images have an average error in final feature positions of 5% to 10% of the vertical distance between eyes and nose, which corresponds to approximately  $2mm$  to  $4mm$ . The standard algorithm performs slightly better than SAFL because  $E_M$  keeps the features fixed to the ground truth positions during part of the optimization. It should be noted that the ground truth positions may have some residual uncertainty, because it is difficult to identify corresponding feature positions (pixel in the image - vertex on the model) exactly by hand. This may explain why the benefit of SAFL in this evaluation criterion becomes visible only on a larger scale of feature distances, i.e. when larger perturbations are applied (Fig. 6, second diagram). These results are consistent with the face identification rates on the left of Fig. 7, where we found similar performance for unperturbed initial features, but a significant improvement for perturbed features.

To demonstrate that SAFL is not restricted to frontal views we did some additional tests on the FERET database. The setting was chosen like for the FRGC data. In a rank 1 identification experiment (1 out of 194) we used *ba* images as gallery and *bb* (rotated views with a mean rotation angle  $\phi$  of  $38.9^\circ$ , cf. 1) as query images also considering perturbation ranges from 0% to 25%. The percentages of correct identification can be found on the right of Fig. 7. Compared to 1) the recognition rates of both (standard and SAFL) are lower. This is due to the fact that in 1) more than the five feature positions are used, e.g. at the ear and at the contour, which are useful for non-frontal views. But our goal here is to demonstrate the usefulness of the new algorithm compared with the standard one.





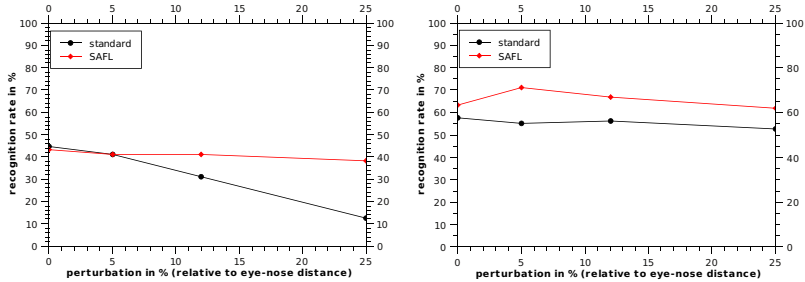
**Fig. 6. Movement of feature positions:** Left diagram: The manually labelled feature positions (without perturbation) were used for initialization of the reconstruction. Right diagram: 25% perturbation. x-axis: Average distance between the manually labelled positions and the resulting positions after reconstruction for a given test image. y-axis: Cumulative error distribution (absolute number of images with distance below threshold, total 300.) Black line: standard 3DMM fitting algorithm, red line: proposed algorithm SAFL.

To test the new algorithm in a *real world scenario* we chose the feature detector of [3] to automatically detect the feature positions on the 300 faces taken from the FRGC database. Performing a rank 1 identification experiment again the standard algorithm delivers a recognition rate of 29.6% and the new algorithm yields a recognition rate of 39.0%. This results are comparable to the recognition rates of the former experiment with random perturbation of 12%.

To confirm our choice of making the features self-adapting and of using the image layer approach rather than considering only the position of maximum feature response, we evaluated some alternative versions of the algorithm:

- a. Standard algorithm, but the initial (perturbed) feature positions (which contribute to  $E_M$ ) are replaced after iteration 1000 by the position of the maximum output of template matching (TM). The idea is that a single TM early in the process would be enough to refine the perturbed feature positions. The template is created after a coarse estimation of pose, lighting and appearance.
- b. Use self-adapting templates that are adapted every 1000 iterations, but consider only the maximum TM output rather than layers  $E_{F_i}$ , and use it in  $E_M$  instead of the initial features (standard algorithm). This condition tests whether the layer approach  $E_{F_i}$  is superior to  $E_M$  which just pulls features to the positions of maximum TM output.
- c. Compute the cross correlation results only once for each feature, and use this to get  $E_{F_i}(x_{F_i}, y_{F_i})$  for the rest of the reconstruction algorithm. This condition verifies the benefit of adaptiveness in the SAFL algorithm.

Table I shows the recognition rates of the standard algorithm, the proposed SAFL and the additional tested versions a,b and c listed above. The results



**Fig. 7. Comparison of recognition rates:** measured in terms of correct identifications (one out of  $n$  individuals). The left diagram is for frontal faces of the FRGC database ( $n = 100$ ) and the right diagram is for non-frontal faces using images from the FERET database *ba* as gallery and *bb* as query images ( $n = 194$ ). The accuracy in the right plot is better than in the left plot because FERET is easier to classify (the images are from a single session).

**Table 1. Recognition rates:** percentage of correct identifications for all algorithms tested on all perturbation ranges

perturbation in %	0	5	12	25
standard	44.6	41.0	31.0	12.3
a	25.0	25.3	25.0	19.6
b	14.0	14.3	11.3	11.3
c	42.0	40.6	39.0	33.0
SAFL	43.3	41.0	41.0	38.3

**Table 2. Computation times:** in seconds, measured on an Intel<sup>R</sup> Core<sup>TM</sup>2 Duo CPU E8300 @ 2.83GHz (single threaded)

facial region	std.	a	b	c	SAFL
541 <sup>2</sup> px	64	92	103	146	160
1054 <sup>2</sup> px	67	120	232	331	456

of the versions (a) and (b) are much lower than all others, indicating that the cost function  $E_{F_i}$  performs better than searching for the maximum output of TM only. The recognition rates of setting (c) come close to the ones of the SAFL approach, but are still inferior, showing that self adaptation is useful. We conclude that both main ideas proposed in this paper make a significant contribution to the stability of 3DMM fitting in a recognition scenario with partially unreliable initial features.

The computation times for the different approaches according to different facial region sizes can be found in Tab. 2. When TM is used, computation times depend critically on the size of the facial region. It would have been worth considering to perform template matching at a lower image resolution.



**Fig. 8. Reconstruction examples:** Each row shows 3DMM fitting for one test image. In the examples in row 1, 2, 3, 4, we used 0%, 5%, 12% and 25% perturbation, respectively, relative to eye-nose distance. From left to right: positions marked on the input image, close-ups of the perturbed feature positions (green: manually labelled, red: perturbed position used), reconstruction with the standard algorithm, reconstruction with the new SAFL approach.

Fig. 8 shows 4 reconstruction results of frontal views. For lack of space, we show only one perturbation level per example in this figure in order to give an idea of what the reconstructions look like but more results can be found in the supplementary material. In the left column, the feature positions are marked on the input images with colored crosses: green for manually labelled positions and red for perturbed positions. The second column shows close-ups of the features randomly chosen for perturbation. In the first row, the manually labelled feature positions were used. Here the SAFL approach got into a local minimum, moving the eyes to the eyebrow positions. In the second row, two randomly chosen feature positions were perturbed 5%. Here the perturbation is quite small, but SAFL outperforms the standard algorithm in reconstruction. In the third row, two randomly chosen feature positions were perturbed 12%. Here it can be seen how much the perturbed feature positions influence the standard algorithm. The reconstruction using SAFL is plausible. In the fourth row, two randomly chosen feature positions were perturbed 25%. We would like to add that both



**Fig. 9. Reconstruction examples of rotated views:** Examples were chosen randomly out of the reconstruction results with a perturbation range of 12%. Each pair of images shows reconstructions using the standard algorithm (left) and SAFL (right).

algorithms may produce suboptimal results occasionally, and we selected four typical examples here.

Results of non-frontal views are shown in Fig. 9. We show only this randomly chosen examples with a perturbation range of 12% in this figure in order to give an idea of what the reconstructions look like but more results can also be found in the supplementary material. At the upper line the SAFL approach improved the reconstruction. On the left side it is obvious but on the right side it can only be seen at the forehead and at the chin. At the lower line the SAFL approach does not really improve the reconstruction. On the left side the model fits better to the image (it is rotated) but it is still too small and on the right side it fits better at the forehead and at the ear but chin and nose are deformed.

## 6 Conclusion

We have presented a new approach for using feature detectors in 3DMM fitting. The algorithm involves adaptive features, which is crucial to leverage the advantages of 3DMMs, and it is based on a new type of cost function that forms a tradeoff between feature similarity and some more global criteria such as geometric configuration, correct reproduction of color values and high prior probability.

The evaluation is based on a scenario where the 3DMM fitting is initialized by a set of potentially unreliable feature detectors, and the algorithm iteratively refines the feature positions. The results indicate that the proposed algorithm improves recognition rates significantly. The second part of our evaluation is focused on the contributions of different design options in our algorithm, and it demonstrates that both the adaptiveness and the new type of cost function increase the robustness.

## References

1. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25, 1063–1074 (2003)
2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: *Computer Graphics Proc. SIGGRAPH'99*, pp. 187–194 (1999)
3. Breuer, P., Kim, K.I., Kienzle, W., Schölkopf, B., Blanz, V.: Automatic 3d face reconstruction from single images or video. In: *FG'08* (2008)
4. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *CVPR'01* (2001)
5. Erukhimov, V., Lee, K.C.: A bottom-up framework for robust facial feature detection. In: *FG'08* (2008)
6. Nguyen, M.H., Perez, J., la Torre Frade, F.D.: Facial feature detection with optimal pixel reduction svms. In: *FG'08* (2008)
7. Zuo, F., de With, P.H.: Facial feature extraction using a cascade of model-based algorithms. In: *AVSBS'05* (2005)
8. Tang, F., Wang, J., Tao, H., Peng, Q.: Probabilistic hierarchical face model for feature localization. In: *WACV'07* (2007)
9. Wimmer, M., Mayer, C., Radig, B.: Robustly classifying facial components using a set of adjusted pixel features. In: *FG'08* (2008)
10. Ardizzone, E., Cascia, M.L., Morana, M.: Probabilistic corner detection for facial feature extraction. In: Foggia, P., Sansone, C., Vento, M. (eds.) *Image Analysis and Processing – ICIAP 2009*. LNCS, vol. 5716, pp. 461–470. Springer, Heidelberg (2009)
11. Kozakaya, T., Shibata, T., Yuasa, M., Yamaguchi, O.: Facial feature localization using weighted vector concentration approach. In: *FG'08* (2008)
12. Oh, J.S., Kim, D.W., Kim, J.T., Yoon, Y.I., Choi, J.S.: Facial component detection for efficient facial characteristic point extraction. In: Kamel, M.S., Campilho, A.C. (eds.) *ICIAR 2005*. LNCS, vol. 3656, pp. 1125–1132. Springer, Heidelberg (2005)
13. Kim, D.H., Dahyot, R.: Face components detection using surf descriptors and svms. In: *IMVIP'08* (2008)
14. Celiktutan, O., Akakin, H.C., Sankur, B.: Multi-attribute robust facial feature localization. In: *FG'08* (2008)
15. Milborrow, S., Nicolls, F.: Locating facial features with an extended active shape model. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 504–513. Springer, Heidelberg (2008)
16. Zuo, F., de With, P.: Fast facial feature extraction using a deformable shape model with haar-wavelet based local texture attributes. In: *ICIP'04* (2004)
17. Cristinacce, D., Cootes, T.: Feature detection and tracking with constrained local models. In: *BMVC'06* (2006)
18. Huang, J., Heisele, B., Blanz, V.: Component-based face recognition with 3d morphable models. In: *Proc. of the 4th Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, Surrey, UK (2003)
19. Gu, L., Kanade, T.: 3d alignment of face in a single image. In: *CVPR'06* (2006)
20. Romdhani, S., Vetter, T.: 3d probabilistic feature point model for object detection and recognition. In: *CVPR'07* (2007)
21. Romdhani, S., Vetter, T.: Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: *CVPR'05* (2005)
22. Burger, W., Burge, M.J.: *Digital Image Processing*. Springer, New York (2008), <http://www.imagingbook.com>
23. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: *CVPR'05* (2005)

# Face Recognition with Patterns of Oriented Edge Magnitudes

Ngoc-Son Vu<sup>1,2</sup> and Alice Caplier<sup>2</sup>

<sup>1</sup> Vesalis Sarl, Clermont Ferrand, France

<sup>2</sup> Gipsa-lab, Grenoble INP, France

**Abstract.** This paper addresses the question of computationally inexpensive yet discriminative and robust feature sets for real-world face recognition. The proposed descriptor named Patterns of Oriented Edge Magnitudes (POEM) has desirable properties: POEM (1) is an oriented, spatial multi-resolution descriptor capturing rich information about the original image; (2) is a multi-scale self-similarity based structure that results in robustness to exterior variations; and (3) is of low complexity and is therefore practical for real-time applications. Briefly speaking, for every pixel, the POEM feature is built by applying a self-similarity based structure on oriented magnitudes, calculated by accumulating a local histogram of gradient orientations over all pixels of image cells, centered on the considered pixel. The robustness and discriminative power of the POEM descriptor is evaluated for face recognition on both constrained (FERET) and unconstrained (LFW) datasets. Experimental results show that our algorithm achieves better performance than the state-of-the-art representations. More impressively, the computational cost of extracting the POEM descriptor is so low that it runs around 20 times faster than just the first step of the methods based upon Gabor filters. Moreover, its data storage requirements are 13 and 27 times smaller than those of the LGBP (Local Gabor Binary Patterns) and HGPP (Histogram of Gabor Phase Patterns) descriptors respectively.

## 1 Introduction

Good pattern representation is one of key issues for all pattern recognition systems. In face recognition, a good representation is one which minimizes intra-person dissimilarities whilst enlarging the margin between different people. This is a critical issue, as variations of pose, illumination, age and expression can be larger than variations of identity in the original face images. For real-world face recognition systems we also believe that a good representation should be both fast and compact: if one is testing a probe face against a large database of desirable (or undesirable) target faces, the extraction and storage of the face representation has to be fast enough for any results to be delivered to the end user in good time. In this paper, we propose a novel feature descriptor named Patterns of Oriented Edge Magnitudes (POEM) for robust face recognition, a descriptor which we argue satisfies these criteria. Experimental results on both

FERET and LFW databases show that POEM method achieves comparable and better performance when compared with state-of-the-art representations. More impressively, the runtime required to extract our descriptor is around 20 times faster than that of even the first step of methods based upon Gabor filters.

We briefly discuss related work in Section 2, describe our method in Section 3. Section 4 details the use of POEM for face recognition. Experimental results are presented in Section 5 and conclusions are given in Section 6.

## 2 Related Work

There is an extensive literature on local descriptors and face recognition. We refer readers to [1, 2] for an in-depth survey, and describe here those high-performing algorithms which are most relevant to our work [3–8].

Local descriptors [1, 7, 9] are commonly employed for many real-world applications because they can be computed efficiently, are resistant to partial occlusion, and are relatively insensitive to changes in viewpoint. Mikolajczyk and Schmid [1] recently evaluated a variety of local descriptors and identified the SIFT (Scale-invariant feature transform) [9] algorithm as being the most resistant to common image deformations. As a dense version of the dominating SIFT feature, HOG [6] has shown great success in object detection and recognition [6, 10] although has not seen much use in face recognition.

For the specific problem of face recognition, there are also many representational approaches, including subspace based holistic features and local appearance features. Heisele *et al.* [11] compared local and global approaches and observed that local systems outperformed global systems for recognition rates larger than 60%. Due to increasing interest, in recent surveys stand-alone sections have been specifically devoted to local methods.

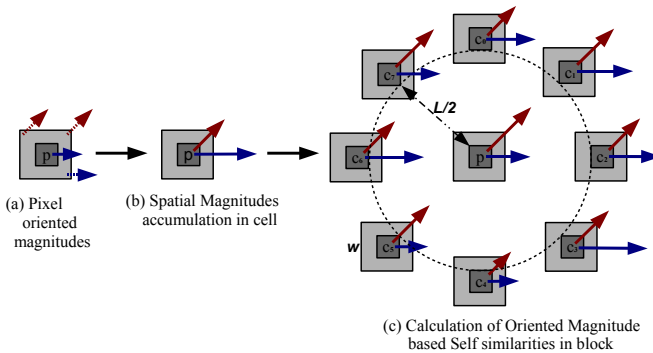
Two of the most successful local face representations are Gabor features [3, 4, 8, 12–15] and Local Binary Patterns (LBP) [5, 13, 16, 17]. Gabor filters, which are spatially localized and selective to spatial orientations and scales, are comparable to the receptive fields of simple cells in the mammalian visual cortex [8]. Due to their robustness to local distortions, Gabor features have been successfully applied to face recognition. Both the FERET evaluation and the FVC2004 contests have seen top performance from Gabor feature based methods. Recently, Pinto *et al.* [14, 15] use V1-like and V1-like+, the Gabor filter based features, as face representation and report good recognition performance on the unconstrained LFW set. Gabor features are typically calculated by convolving images with a family of Gabor kernels at different scales and orientations, which is a costly stage. Despite recent attempts at speeding this process up (e.g. the Simplified Gabor Wavelets of Choi *et al.* [18]) the process of extracting these features is still prohibitive for real-time applications.

More recently, the spatial histogram model of LBP has been proposed to represent visual objects, and successfully applied to texture analysis [19], and face recognition [5]. LBP is basically a fine-scale descriptor that captures small texture details, in contrast to Gabor features which encode facial shape and

appearance over a range of coarser scales. By using LBP, Ahonen *et al.* [5] have reported impressive results on the FERET database. Several variants of LBP are also presented and successfully applied to different applications, such as Center-Symmetric LBP (CSLBP) [20], Three-Patch LBP (TPLBP), Four-Patch LBP (FPLBP) [16], etc.

A combination approach was introduced by Zhang *et al.* [3] extending LBP to LGBP by introducing multi-orientation and multi-scale Gabor filtering before using LBP and impressively improved the performance when compared with pure LBP. In a similar vein, they further proposed HGPP [4] combining the spatial histogram and the Gabor phase information encoding scheme. In [13], a model fusing the multiple descriptor sets is presented with very high performance on constrained datasets. More recently, Wolf *et al.* [17] combine LBP, TPLBP, FPLBP, Gabor and SIFT with different similarity measures, showing promising results on the LFW set.

These combination methods try to bring the advantages of LBP and Gabor filters together, but they also bring the disadvantages of Gabor based systems; namely computational cost and storage requirements.



**Fig. 1.** Main steps of POEM feature extraction

The aim of this study is to find a feature descriptor that can inherit various good properties from existing features but with low computational cost. We propose applying the LBP based structure on oriented magnitudes to build a novel descriptor: Patterns of Oriented Edge Magnitudes (POEM). Briefly speaking, in order to calculate the POEM for one pixel, the intensity values in the calculation of the traditional LBP are replaced by the gradient magnitudes, calculated by accumulating a local histogram of gradient directions over all pixels of a spatial patch (“cell”). Additionally, these calculations are done across different orientations. We use the terms *cell* and *block*, as in [6], but with a slightly different meaning. Cells (big squares in Figure 1a) refer to spatial regions around the current pixel where a histogram of orientation is accumulated and assigned to the cell central pixel. Blocks (circular in Figure 1c) refer to more extended



spatial regions, on which the LBP operator is applied. Note that our use of oriented magnitudes is also different from that in [6] where HOG is computed in dense grids and then is used as the representation of cell. On the contrary, in POEM, for *each pixel*, a local histogram of gradient over all pixels of cell, centered on the considered pixel, is used as *the representation of that pixel*. Similarly, the term *pattern* in POEM is not as *local* as in the conventional LBP based methods. LBP methods often calculate the self-similarity within a small neighborhood while the block used in POEM is rather extended (see details in sections 3.2 and 5.1).

In combination approach [3], Gabor filters are first used for capturing large scale information and LBP operator is then applied for encoding the small details. On the contrary, POEM first characterizes object details in small scale and then uses the LBP based structure to encode information over larger region.

### 3 POEM Descriptor

Similar features have seen increasing use over the past decade [6, 7, 9]; the fundamental idea being to characterize the local object appearance and shape by the distribution of local intensity gradients or edge directions. We further apply the idea of self-similarity calculation from LBP-based structure on these distributions since we find that combining both the edge/local shape information and the relation between the information in neighboring cells can better characterize object appearance. As can be seen in Figure 1, once the gradient image is computed, the next two steps are assigning the cell's accumulated magnitudes to its central pixel, and then calculating the block self-similarities based on the accumulated gradient magnitudes by applying the LBP operator.

#### 3.1 POEM Feature Extraction in Detail

The first step in extracting the POEM feature is the computation of the gradient image. The gradient orientation of each pixel is then evenly discretized over  $0-\pi$  (*unsigned* representation) or  $0-2\pi$  (*signed* representation). Thus, at each pixel, the gradient is a 2D vector with its original magnitude and its discretized direction (the blue continuous arrow emitting from pixel **p** in Figure 1a).

The second step is to incorporate gradient information from neighbouring pixels (the discontinuous arrows in Figure 1a) by computing a local histogram of gradient orientations over all cell pixels. Vote weights can either be the gradient magnitude itself, or some function of the magnitude: we use the gradient magnitude at the pixel, as in [6]. At each pixel, the feature is now a vector of  $m$  values where  $m$  is the number of discretized orientations (number of bins).

Finally, we encode the accumulated magnitudes using the LBP operator within a block. The original LBP operator labels the pixels of an image by thresholding the  $3 \times 3$  neighborhood surrounding the pixel with the intensity value of central pixel, and considering the sequence of 8 result bits as a number (as shown in Figure 2). Only uniform patterns, which are those binary patterns that have at most 2 transitions from 0 to 1, are typically used to accelerate the method.

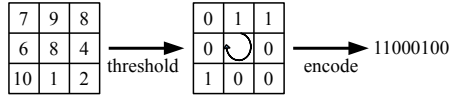


Fig. 2. LBP operator

We apply this procedure on the accumulated gradient magnitudes and across different directions to build the POEM. Firstly, at the pixel position  $p$ , a POEM feature is calculated for each discretized direction  $\theta_i$ :

$$POEM_{L,w,n}^{\theta_i}(p) = \sum_{j=1}^n f(S(I_p^{\theta_i}, I_{c_j}^{\theta_i}))2^j, \tag{1}$$

where  $I_p, I_{c_j}$  are the accumulated gradient magnitudes of central and surrounding pixels  $p, c_j$  respectively;  $S(.,.)$  is the similarity function (e.g. the difference of two gradient magnitudes);  $L, w$  refer to the size of blocks and cells, respectively;  $n$ , set to 8 by default in this paper, is number of pixels surrounding the considered pixel  $p$ ; and  $f$  is defined as:

$$f(x) = \begin{cases} 1 & \text{if } x \geq \tau, \\ 0 & \text{if } x < \tau, \end{cases} \tag{2}$$

where the value  $\tau$  is slightly larger than zero to provide some stability in uniform regions, similar to [16].

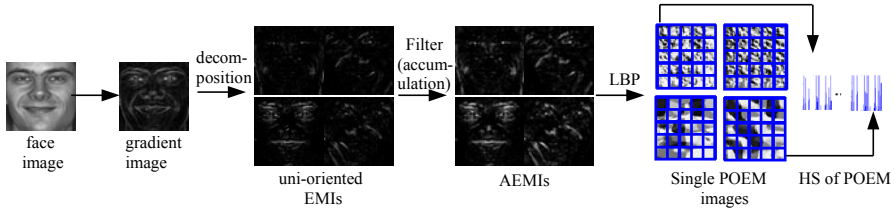
The final POEM feature set at each pixel is the concatenation of these unidirectional POEMs at each of our  $m$  orientations:

$$POEM_{L,w,n}(p) = \{POEM^{\theta_1}, \dots, POEM^{\theta_m}\}, \tag{3}$$

### 3.2 Properties of POEM

We discuss here the good properties of this novel descriptor for object representation postponing the question of complexity until Section 5.4. For each pixel, POEM characterizes not only local object appearance and shape, but also the relationships between this information in neighboring regions. It has the following properties:

- POEM is an oriented feature. Since the number of discretized directions can be varied, POEM has the ability to capture image information in any direction and is adaptable for object representation with different levels of orientation accuracy.
- Computed at different scales of cells and blocks, POEM is also a spatial multi-resolution feature. This enables it to capture both local information and more global structure.
- Using gradient magnitudes instead of the pixel intensity values for the construction makes POEM robust to lighting variance. In [21, 22], edge magnitudes have been shown to be largely insensitive to lighting.



**Fig. 3.** Implementation of POEM for face description

- The oriented magnitude based representation contains itself the relation between cell pixels. POEM further calculates dissimilarities between cells and therefore has the ability to capture multi-scale self-similarities between image regions. This makes POEM robust to exterior variations, such as local image transformations due to variations of pose, lighting, expression and occlusion that we frequently find when dealing with faces.

Patch-based or multi-block LBP [16] also considers relationships between regions in a similar way to our POEM descriptor. However the richer information coming from the use of gradients at multiple orientations gives us greater descriptive power, and a greater insensitivity to lighting variations.

## 4 Face Recognition Based on POEM

For face recognition, we use a similar procedure to that described in [5], except that each pixel is characterized with the POEM features instead of a LBP code (cf., Figure 3).

### POEM Histogram Sequences for Face Recognition

In practice, the Oriented Edge Magnitude Image (oriented EMI) is first calculated from the original input image (section 3.1) and divided into  $m$  uni-oriented EMIs through gradient orientations of pixels. Note that the pixel value in uni-oriented EMIs is gradient magnitude. For every pixel on uni-oriented EMIs, its value is then replaced by the sum of all values in the cell, centered on the current pixel. These calculations are very fast (using the advantage of integral image [23]). Result images are referred to accumulated EMIs (AEMIs). LBP operators are applied on these AEMIs to obtain the POEM images (Figure 3). In order to incorporate more spatial information into the final descriptor, the POEM images are spatially divided into multiple non-overlapping regions, and histograms are extracted from each region. Similar to [5, 16], only *uniform* POEM codes are used. Finally, all the histograms estimated from all regions of all POEMs are concatenated into a single histogram sequence (POEM-HS) to represent the given face.

Given two histogram sequences of POEM representing two face images, we use the chi-square distance between histograms [5] to measure the similarity between two images.

## 5 Experiments and Discussions

In this section, we conduct comparison experiments on two face databases, FERET (controlled variations) [24] and LFW (unconstrained environments) [25], in order to validate the efficiency of the proposed descriptor for face recognition.

### 5.1 Parameter Evaluation

In this section we consider how the parameters of POEM influence its final performance. Parameters varied include the number  $m$  and type (unsigned or signed) of orientations, the cell size ( $w * w$ ), and block size ( $L * L$ ). As for the cell/block geometry, two main geometries exist: rectangular and circular. In this paper we use circular blocks including bilinear interpolation for the values since they provide the relation between equidistant neighboring cells [5]. Square cells are used, meaning that pixel information is calculated using its neighborhood in a square patch.

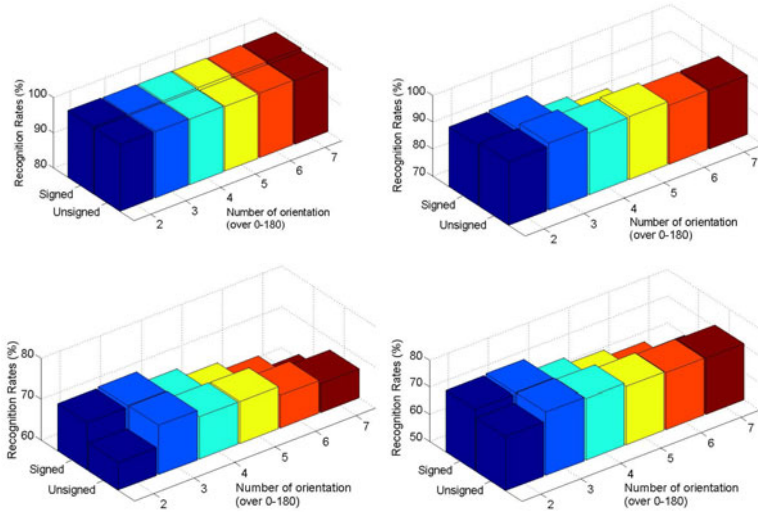
The experiments checking the effects of parameters are conducted on the FERET face database, following the standard evaluation protocol: Fa containing 1196 frontal images of 1196 subjects is used as Gallery, while Fb (1195 images of expression variations), Fc (194 images of illumination variations), Dup I & Dup II(722 & 234 images taken later in time) are the Probe sets.

The facial images of FERET are first cropped and aligned using the given coordinates of two eyes. We roughly fix the width and height of face about two times of distance between the centers of the two eyes, and then resize image to 110x110 pixels (cf., the first image in Figure 3). We do not use any other particular face mask, such as in [5] for example. In our experiments, we divide each image into 10x10 non overlapping patches. Since this paper concentrates on the feature sets, we use a simple nearest neighbor classifier to calculate the recognition rates and consider classifier choice beyond the scope of the current paper. But we believe that better classifier could enhance recognition performance.

**Experiment 1, concerning the number of orientations and signed/unsigned representation.** Nearly six hundred cases are considered, recognition rates are calculated on 3000+ face images with different parameters:  $L = \{5, 6, 7, 8, 9, 10, 11\}$ ,  $w = \{3, 4, 5, 6, 7, 8\}$ , the numbers of discretized orientations are  $m = \{2, 3, 4, 5, 6, 7\}$  in the case of unsigned representation, and are doubled to  $m = \{4, 6, 8, 10, 12, 14\}$  in the case of signed representation. Cells can overlap, notably when blocks are smaller than cells, meaning that each pixel can contribute more than once.

For each Probe set, the average rates are calculated over different numbers and types of orientation. Figure 4 shows the recognition rates obtained on Probe sets Fb, Fc, Dup1, and Dup2. The average recognition rates obtained on Probe set Fb (as shown in the Figure 4a) are around 96.5%, representing an improvement of about 3.5% in comparison with LBP [5].

Considering the question of using a signed or an unsigned representation, we find similar results to [6], in that including signed gradients decreases the



**Fig. 4.** Recognition rates obtained with different numbers of orientations on Probe sets: Fb (a), Fc (b), Dup1 (c), and Dup2 (d). These rates are calculated by averaging recognition rates with different sizes of cell/block.

performance of POEM even when the data dimension is doubled to preserve more original orientation resolution. For face recognition, POEM provides the best performance with only 3 unsigned bins. This should be noted as one advantage of POEM since the data dimension for face description is not greatly increased as in LGPP or HGPP [3, 4]. It is clear from Figure 4 (c,d) that using too many orientations degrades significantly the recognition rates on Dup1 and Dup2 sets. This can be explained by the fact that increasing the number of orientation bins makes POEM more sensitive to wrinkles appearing in face with time.

Summarizing, the number  $m$  and signed/unsigned orientations do not affect the recognition performance in the case of expression variations (Fb set); the unsigned representation is more robust than signed representation, notably to lighting; using few orientations (1, 2) is not enough to represent face information but too many (more than 3) makes POEM sensitive to aging variations. Thus, the best case is 3 unsigned bins.

**Experiment 2, concerning the size of cells and blocks.** Average recognition rates of all four Probe sets are first calculated with different sizes of cells and blocks with 3 unsigned bins of orientation discretization. As can be seen from Figure 5, using POEM built on 10x10 pixel blocks with histogram of 7x7 pixel cells provides the best performance.

To verify the correctness of these parameters, we further calculate the average rates across cell sizes and across block sizes, meaning that these parameters are now considered independent. Also, in this test, both 10x10 pixel block and 7x7 pixel cell

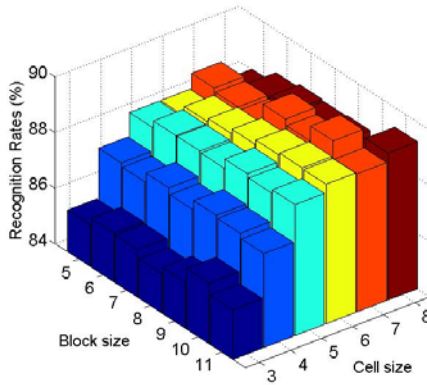


Fig. 5. Recognition rates as the cell and block sizes change

performed the best. This procedure has been repeated with different numbers of orientation bins, and the same optimal parameters have been obtained.

In conclusion, the optimal POEM parameters for face recognition are: unsigned representation with 3 bins, built on 10x10 pixel blocks and 7x7 pixel cells.

## 5.2 Results Based Upon the FERET Evaluation Protocol

We consider the FERET'97 results [24], results of the LBP [5], HGPP [4], LGBPHS [3], and more recent results in [12, 13, 26]. These results, to the best of our knowledge, are the state-of-the-art with respect to the FERET dataset.

As can be seen from Table 1, in comparison with the conventional LBP and HOG (the performance of HOG for face recognition is reported in [26]), our POEM descriptor is much more robust to lighting, expression & aging, illustrated by significant improvements in recognition rates for all probe sets. While compared with LGBP and HGPP, reported as being the best performing descriptors on FERET database, POEM provides comparable performance for the probe sets Fb and Fc. When we consider the more challenging probe sets Dup1 and Dup2, POEM outperforms LGBP and is comparable to HGPP.

Concerning the results of [12, 13], they are only suitable for very limited reference since they are obtained by using a more complex classification phase, and we wish to concentrate upon the performance of the descriptor rather than the classifiers. In [12], Zou *et al.* use Gabor jets as local facial features which are compared using normalized inner products at different scales, and results are combined using the Borda Count method. Moreover, Zou *et al.* do not use only the pure face area, as defined in [27]. In [13], Tan and Triggs fuse two feature sets, Gabor & LBP, and use a complex dimensionality reduction and classification phase, PCA & Kernel DCV. Their method suffers from the disadvantage that adding a new individual to the gallery requires recalculating all existing coefficients: PCA coefficients of Gabor & LBP features, and the KDCV coefficients of the fused features.

**Table 1.** Recognition rate comparisons with other state-of-the-art results tested with Feret evaluation protocol

Methods	Fb	Fc	Dup1	Dup2
LBP [5]	93.0	51.0	61.0	50.0
LGBPHS [3]	94.0	97.0	68.0	53.0
HGPP [4]	97.6	98.9	77.7	76.1
HOG [26]	90.0	74.0	54.0	46.6
<b>POEM</b>	<b>97.6</b>	<b>96</b>	<b>77.8</b>	<b>76.5</b>
<b>Retina filter [28] + POEM</b>	<b>98.1</b>	<b>99</b>	<b>79.6</b>	<b>79.1</b>
<i>Results of [12]</i>	<i>99.5</i>	<i>99.5</i>	<i>85</i>	<i>79.5</i>
<i>Results of [13]</i>	<i>98</i>	<i>98</i>	<i>90</i>	<i>85</i>

We further employ the real-time retina filtering presented by Vu and Caplier in [28] as preprocessing step since this algorithm, as pointed out by authors, not only removes the illumination variations but also enhances the image edges, upon which our POEM is constructed. It is clear from Table 1 that the retina filter enhances the performance of POEM, especially for the probe set Fc.

### 5.3 Results on LFW Dataset

In order to test the performance of the POEM descriptor across different databases, we duplicate these experiments on another well-known dataset, LFW [25], containing 13233 face images of 5749 individuals. This database is described as “unconstrained”, meaning that face images are subject to a large range of “natural” variations. The operational goal of this set differs from above FERET database; it is aimed at studying the problem of face pair matching (given two face images, decide whether they are from the same person or not). We follow the standard procedure described in [25] and report the mean classification accuracy  $\pm$  standard error computed from 10 folds of the “Image-Restricted View 2” portion of LFW set.

**Fig. 6.** Examples of LFW images used in our tests

As mentioned above, the goal of the current paper is to demonstrate the efficiency of the novel descriptor, not to compete in the LFW challenge. We therefore report the obtained results using POEM descriptor in a simple threshold-on-descriptor-distance classification context [16], meaning that for each test fold, an optimal threshold giving the highest separation score on the 5400 examples of the training set is chosen and then is used to calculate the classification accuracy for the 600 examples of the test set. We only compare our results with other

**Table 2.** Recognition results of different methods on LFW set, Image-Restricted Training, View 2

Reference	Descriptors (similarity measure)	Performance
Pinto2008 [14]	V1-like	0.6421 $\pm$ 0.0069
	V1-like+	0.6808 $\pm$ 0.0045
Wolf2009 [17]	LBP Euclidean/SQRT	0.6824/0.6790
	Gabor Euclidean/SQRT	0.6849/0.6841
	TPLBP Euclidean/SQRT	0.6926/0.6897
	FPLBP Euclidean/SQRT	0.6818/0.6746
	SIFT Euclidean/SQRT	0.6912/0.6986
	All combined	0.7521 $\pm$ 0.0055
<b>This paper</b>	<b>POEM</b>	<b>0.7400 <math>\pm</math> 0.0062</b>
	<b>POEM Flip</b>	<b>0.7542 <math>\pm</math> 0.0071</b>

descriptor-based results and refer readers to [29] for further algorithm classifiers reported on the LFW dataset.

In our experiments, the LFW gray images aligned automatically by Wolf *et al.* [17] are used and cropped to 100 x 116 pixels around their center. As is clear from the Figure 6, there is significant pose variation within this dataset. In order to address this we flip image 1 of each pair on the vertical axis, and take the smaller of the two histogram distances as our measure. This simple preprocessing step improves recognition rates and is referred to as POEM-Flip in following results. Because of the poor quality of the images in the LFW dataset, retina filtering does not improve the recognition results. With low quality images, the retina filter enhances the image contours and removes illumination variations but also enhances image artifacts (such as those arising from compression). The similar performance is obtained in both cases that the retina filter is used or not. Otherwise, we do not employ any other preprocessing technique.

It is clear from Table 2 that POEM method outperforms all other competing descriptors: LBP, TPLBP, FPLBP, Gabor filters and SIFT. When compared with these descriptors, the POEM based method represents around 20% reduction in classification error. Our POEM-flip mean recognition rate 75.42% is better than that of the “combined” method of [17]. It is worth noting that Wolf *et al.* [17] combine 10 descriptor/mode scores using SVM classification. The results in [14], based upon Gabor filters, are much worse than ours.

#### 5.4 A Consideration of Computational Cost

In this section we compare the complexity of POEM with two of the most widely used descriptors for face recognition: LBP and Gabor wavelet based methods. Considering the pure one-LBP-operator method, POEM based face recognition requires a computational complexity which is 3 times higher (the calculation of integral gradient image is very fast when compared to the calculation of POEM features and the construction of POEM-HS) but at the same time, there are



**Table 3.** Runtime required to extract the whole POEM descriptor and the initial step of Gabor based feature extraction. Calculated using the implementation in Matlab, these times are only suitable for rough comparisons of computational complexity.

Methods	Times (seconds)
Convolution with 40 Gabor kernels	0.4349
POEM extraction	0.0191

remarkable improvements in recognition rates on the FERET database (+5%, +45%, +16.6% and +26.5% for the probe sets Fb, Fc, Dup1 and Dup2, respectively). And on the LFW set, POEM method also outperforms other variants of LBP, TPLBP and FPLBP.

When we consider Gabor filter based descriptors, only the runtime required for the convolution of the image with the family of Gabor kernels (8 orientations and 5 scales) is necessary. From Table 3, we see that the computation of the whole POEM descriptor is about 23 times faster than that of just this first step of Gabor feature extraction.

We do not calculate here the time required to extract SIFT descriptor and do not compare directly it to POEM, but as argued in [20], SIFT is about 3 times slower than  $3 \times 3$  grid Center-Symmetric LBP (CSLBP), a variant of LBP ( $3 \times 3$  grid CSLBP means that the descriptor is obtained concatenating the histogram of CSLBP features over grid of  $3 \times 3$ ). Thus it seems that POEM and SIFT have the similar time complexity. However, for face recognition, POEM clearly outperforms SIFT, representing about 20% reduction in classification error on LFW set. Retina filtering is a linear and real-time algorithm. Its calculation time is about 1/5 of that required to extract POEM.

Considering data storage requirements, for a single face, the size of a complete set of POEM descriptors is 13 and 27 times smaller than that of LGBP and HGPP (LGBP calculates LBP on 40 convolved images while HGPP encodes both real and imaginary images). Note that these comparisons are roughly done considering all 256 patterns of our POEM features. However, in this paper, we use only 59 uniform POEMs, meaning that the size of POEM descriptors used here is 58 and 116 times smaller than that of LGBP and HGPP (these ones use all 256 feature values). When compared to the “combined” method of Wolf *et al.* [17], the space complexity of POEM descriptor is considerably smaller. For one patch, the size of POEM-HS is  $59 \times 3$ , while the size of method in [17] is  $59 \times 2 + 16$  (the size of LBP, TPLBP, and FPLBP per patch are 59, 59 & 16, respectively) + 128 (dimension of SIFT) + size of Gabor based descriptor (which is equal to the size of patch  $\times$  number of scales  $\times$  number of orientations as in [17, 26]).

Thus we agree that the POEM descriptor is the first to allow high performance real-time face recognition. Low complexity descriptors provide worse results; whilst representations based upon multiple feature types can achieve similarly high performance but are too slow for real-time systems.

## 6 Conclusion

By applying the LBP operator on accumulated edge magnitudes across different directions, we have developed a novel descriptor for face representation which has several desirable features. It is robust to lighting, pose and expression variations, and is fast to compute when compared to many of the competing descriptors. We have shown that it is an effective representation for face recognition in both constrained (FERET) and unconstrained (LFW) face recognition tasks outperforming all other purely descriptor based methods. This high performance coupled with the speed of extraction suggests that this descriptor is a good candidate for use in real-world face recognition systems.

Future work will involve testing the POEM descriptor in a broader range of computer vision tasks, such as face detection and object recognition. We will also investigate the use of more powerful classifiers alongside the POEM descriptor within the face recognition domain.

## References

1. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE TPAMI* 27, 1615–1630 (2005)
2. Zhao, W., Chellappa, R., Corporation, S., Rosenfeld, A., Phillips, P.J.: Face recognition: A literature survey. *ACM Computing Surveys* (2000)
3. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In: *ICCV*, vol. 1, pp. 786–791 (2005)
4. Zhang, B., Shan, S.S., Chen, X.: Gao: Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. *IEEE Trans. on Image Processing* 16, 57–68 (2007)
5. 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)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, Washington, DC, USA, pp. 886–893. IEEE Computer Society, Los Alamitos (2005)
7. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: *CVPR* (2004)
8. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Trans. on Image Processing* 11, 467–476 (2002)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision* 60, 91–110 (2004)
10. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model (2008)
11. Heisele, B., Ho, P., Wu, J., Poggio, T.: Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding* 91, 6–21 (2003)
12. Zou, J., Ji, Q., Nagy, G.: A comparative study of local matching approach for face recognition. *IEEE Trans. on Image Processing* 16(10), 2617–2628 (2007)

13. Tan, X., Triggs, B.: Fusing gabor and lbp feature sets for kernel-based face recognition. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 235–249. Springer, Heidelberg (2007)
14. Pinto, N., DiCarlo, J., Cox, D.: Establishing good benchmarks and baselines for face recognition. In: Faces in Real-Life Images Workshop in ECCV (2008)
15. Pinto, N., DiCarlo, J., Cox, D.: How far can you get a modern face recognition test set using only simple features? In: CVPR (2009)
16. Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: Real-Life Images workshop at ECCV (2008)
17. Wolf, L., Hassner, T., Taigman, Y.: Similarity scores based on background samples. In: ACCV (2009)
18. Choi, W., Tse, S., Wong, K., Lam, K.: Simplified gabor wavelets for human face recognition. *Pattern Recognition* 41, 1186–1199 (2008)
19. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI* 24, 971–987 (2002)
20. Heikkila, M., Pietikainen, M., Schmid, C.: Description of interest regions with local binary patterns. *Pattern Recognition* 42, 432–436 (2009)
21. Chen, H.F., Belhumeur, P.N., Jacobs, D.W.: In: search of illumination invariants. In: CVPR (2000)
22. Ling, H., Soatto, S., Ramanathan, N., Jacobs, D.: A study of face recognition as people age. In: ICCV (2007)
23. Viola, P., Jones, M.: Robust real-time face detection. *Int. Journal of Computer Vision* 57, 137–154 (2004)
24. Phillips, J., Moon, H., Rizvi, S.A., et al.: The feret evaluation methodology for face-recognition algorithms. *IEEE TPAMI* 22, 1090–1104 (2000)
25. Huang, G.B., Manu Ramesh, T.B., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst (2007)
26. Meyers, E., Wolf, L.: Using biologically inspired features for face processing. *Int. Journal of Computer Vision* 76, 93–104 (2008)
27. Chen, L., Liao, H., Lin, J., Han, C.: Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof. *Pattern Recognition* 34(5), 1393–1403 (2001)
28. Vu, N., Caplier, A.: Illumination-robust face recognition using the retina modelling. In: ICIP (2009)
29. LFWresults: <http://vis-www.cs.umass.edu/lfw/results.html>

# Spatial-Temporal Granularity-Tunable Gradients Partition (STGGP) Descriptors for Human Detection

Yazhou Liu, Shiguang Shan, Xilin Chen, Janne Heikkila,  
Wen Gao, and Matti Pietikainen

Key Laboratory of Intelligent Information Processing, Institute of Computing  
Technology

Chinese Academy of Sciences (CAS), China

Machine Vision Group, Department of Electrical and Information Engineering  
University of Oulu, Finland

{yzliu,sgshan,xlchen,wgao}@jdl.ac.cn

{Janne.Heikkila,Matti.Pietikainen}@ee.oulu.fi

**Abstract.** This paper presents a novel descriptor for human detection in video sequence. It is referred to as spatial-temporal granularity-tunable gradients partition (STGGP), which is an extension of granularity-tunable gradients partition (GGP) from the still image domain to the spatial-temporal domain. Specifically, the moving human body is considered as a 3-dimensional entity in the spatial-temporal domain. Then in 3D Hough space, we define the generalized plane as a primitive to parse the structure of this 3D entity. The advantage of the generalized plane is that it can tolerate imperfect planes with certain level of uncertainty in rotation and translation. The robustness to the uncertainty is controlled quantitatively by the granularity parameters defined explicitly in the generalized plane. This property endows the STGGP descriptors versatile ability to represent both the deterministic structures and the statistical summarizations of the object. Moreover, the STGGP descriptor encodes much heterogeneous information such as the gradients' strength, position, and distribution, as well as their temporal motion to enrich its representation ability. We evaluate the STGGP on human detection in sequence on the public datasets and very promising results have been achieved.

## 1 Introduction

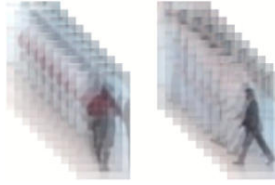
Human detection research has received more and more attention in recent years because of increasing demands in practical applications, such as smart surveillance system, on-board driving assistance system and content based image/video management system. Even through remarkable progress has been achieved [1,2,3,4,5,6,7], finding the human is still considered as one of the hardest task for object detection. The difficulties come from the articulation of human body, the inconsistency of clothes, the variation of the illumination and the unpredictability of the occlusion.

Human detection from the still images has been one of the most active research fields during the recent years. Varieties of features have been invented to overcome the difficulties mentioned above. Earlier works for human detection started from Haar-like features, which have been applied to face detection task successfully [8,9,10]. Because of the large variation of human clothes and background, some researchers turned to the contour based descriptors. Gavrilu [11] presented a contour based hierarchical chamfer matching detector. Lin et al. [12,13] extended this work by decomposing the global shape models into parts to construct a parts template based hierarchical tree. Ferrari et al. [14] used the network of contour segments to represent the shape of the object. Wu and Nevatia [15] used edgelet to represent the local silhouette of the human.

After the invention of the SIFT descriptor [16], more researchers have used the statistical summarization of the gradients to represent human body. Such as the position-orientation histogram features proposed by Mikolajczyk et al. [17]; the histograms of oriented gradients (HOG) proposed by Dalal et al. [18,19] and it's improvements [20]; the covariance matrix descriptor proposed by Tuzel et al. [21]; and the HOG-LBP descriptor proposed by Wang et al. [2]. More recently, granularity-tunable gradients partition (GGP) for human detection was proposed by Liu et al. [22], in which granularity is used to define the spatial and angular uncertainty of the line segments in the Hough space. By adjusting the granularity, GGP provides a container of descriptors from deterministic to statistic.

Even with these powerful representation methods, the appearance of human body is still not discriminative enough, especially in some complex environments. Therefore, some works use the motion information to improve the performance of human detection. As mentioned in [23], certain kinds of movement are characteristics of humans, so detector performance can potentially be improved by including motion information. Viola et al. [24] used the Haar-like filters to extract the appearance from the single image and extract the motion information from the consecutive frames. By including the motion information, they can improve the performance of their system remarkably. Dalal et al. [23] used orientated histograms of differential optical flow to capture the motion information of the human, and then they combined the motion descriptors with histogram of oriented gradient appearance descriptors. The combined detector can reduce the false alarm rate by a factor of 10. Similar improvements have also been reported by Wojek et al. in their resent work [25].

These works show that incorporating the motion and appearance information is a promising way to improve the performance of human detection. Therefore, this work extends the granularity-tunable gradients partition (GGP) [22] from the image domain to the spatial-temporal domain. This new descriptor is referred to as spatial-temporal granularity-tunable gradients partition, or STGGP for short. In STGGP, human and their motions are modeled in the joint spatial-temporal domain. The spatial-temporal volume representations has been widely used in the action recognition research, as in [26,27,28], and very promising results have been reported. But for human detection research, most of the well-



**Fig. 1.** The spatial-temporal representation of human body

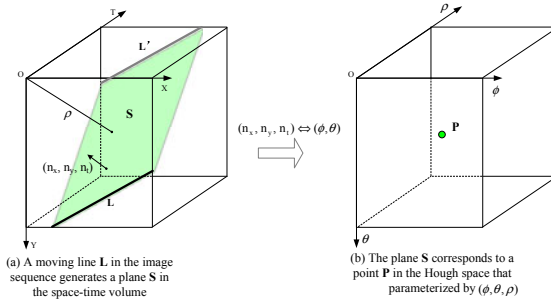
known methods, such as [24,23,25], model the human appearance and motion information as two separate channels. This work considers the moving human body as a 3-dimensional entity in the spatial-temporal domain. Then we use the *generalized planes* to parse the structure of this 3D entity.

The *generalized planes* are defined in the Hough space, which extend the representation of the plane from a point to a cuboid region. The size of the cuboid region is related to a certain level of robustness to rotation and translation uncertainty. Therefore, by changing the size of the cuboid region, the robustness can be controlled explicitly. Hence, a family of descriptors with different representation abilities can be generated that range from the specific geometrical representation to the statistical summarization of the object. This multiple representation property is referred to as *granularity-tunability* and the size parameters of the cuboid region is referred to as *granularity parameters*, or *granularity* for short. This property enables the STGGP descriptor to represent the complex human pattern in the spatial-temporal domain.

The rest of the paper is organized as follows: Section 2 introduces the human representation method in the spatial-temporal domain; Section 3 defines the generalized plane; Section 4 presents the mapping method of the generalized plane from the Hough space to the spatial-temporal domain; Section 5 gives the computational details; and Section 6 contains the experimental results.

## 2 Spatial-Temporal Volume Representation of Human Body in Video

The spatial-temporal volume (STV) is used as one of the basic representations of the human body in video, refer to Fig. 1 for example. This volume contains two image axes  $X$  and  $Y$ , and a temporal axis  $T$ , therefore it can encode both the appearance and motion information of the human body. Unlike the previous works [24,23] which extract the appearance and motion information as two separate channels, in this work, the moving human body is considered as a 3D entity in the spatial-temporal domain. This 3D entity comes from the motion of the contours/edges. When a contour/edge in the image plane moves along the temporal axis, its trajectory will extend a surface in the spatial-temporal domain. Take Fig. 2 for example, the line  $L$  in frame  $I_0$  translates to the line  $L'$  in frame  $I_{T-1}$  through a uniform linear motion. Its trajectory from frame



**Fig. 2.** The 3D planes that generated by the human motion and its mapping in the Hough space

$I_0$  to  $I_{T-1}$  can expand a plane  $S$ . When the contour/edge is not linear or the motion is not uniform, the plane will change to a surface, called by us the spatial-temporal surface. Therefore, the moving human body can be considered as the combination of many spatial-temporal surfaces.

There are two challenges for this surface based human representation: firstly, since the contours/edges in the real-world images are usually not well defined geometrical structures and the motions of the human body are usually complex, the spatial-temporal surfaces may not be in any well defined geometrical forms and can not be explained analytically; secondly, due to the imperfections in either the image data or the edge detector, the contours/edges in the images may not be smooth and continuous. Therefore the smoothness and continuity of the spatial-temporal surfaces can not be guaranteed.

For the first challenge, a possible solution is to use the combination of smaller 3D facets to approximate the surfaces with arbitrary structure. In this way, the 3D planes (facets) are further introduced as the primitives to represent the spatial-temporal surface, and the moving human body can be parsed as a combination of these planes. Regarding the second challenge, we extend the definition of the plane to make it to tolerate the discontinuity using spatial and angular uncertainty. This relaxed definition of the plane is referred to as *generalized plane*, in which the uncertainty of the rotation and translation are defined explicitly. More details will be presented in the following sections.

### 3 The Definition of the Generalized Plane

In the 3D spatial-temporal domain, a plane is represented by its explicit equation as  $t = ax + by + c$ . Here, we can use a 3D Hough space corresponding to the parameters  $a$ ,  $b$  and  $c$ . However, this formulation suffers from the following problem: as the planar direction becomes vertical, the values of some parameters will become too big and even infinite. This means some planes are not well defined in this  $a - b - c$  Hough space.

To avoid the above problem, we parameterize the plane by its normal direction  $\mathbf{n} = (n_x, n_y, n_t)$  and its perpendicular distance  $\rho$  from the origin instead, as in Fig.2(a). This is also called Hesse normal form of the plane, and can be represented as follows:

$$\rho = \mathbf{p} \cdot \mathbf{n} \quad (1)$$

where  $\mathbf{p} = (x, y, t)$  is the coordinates of the points on the plane. As there is a constraint on the magnitude of the normal of the plane, i.e.  $\|\mathbf{n}\| = 1$ , there are only two degrees of freedom for  $\mathbf{n} = (n_x, n_y, n_t)$ . Therefore, the normal direction  $\mathbf{n}$  can be represented by the spherical coordinates of a unit sphere  $(\phi, \theta)$  as:

$$\mathbf{n} = (\sin \phi \cos \theta, \sin \phi \sin \theta, \cos \phi) \quad (2)$$

where the inclination  $\phi \in [0, \pi]$  is the angle between the zenith direction and  $\mathbf{n}$ ; the azimuth  $\theta \in [0, 2\pi)$  is the angle between the reference direction on the chosen plane and the projection of  $\mathbf{n}$  on the plane, as shown in Fig.3(b).

Therefore, by replacing the Equ.2 into the Equ.1, we can get the representation of the plane in the spherical coordinates as:

$$\rho = x \sin \phi \cos \theta + y \sin \phi \sin \theta + t \cos \phi \quad (3)$$

In this definition, there are three parameters  $\phi$ ,  $\theta$  and  $\rho$ , and the Hough space can be defined accordingly. We refer to this Hough space as the  $\phi - \theta - \rho$  Hough space and all the planes can be well defined in this space. Any plane  $S$  in the STV space can map to a point  $P$  in this Hough space, as shown in Fig.2(b). Any point  $(x, y, t)$  on this plane satisfies the definition:

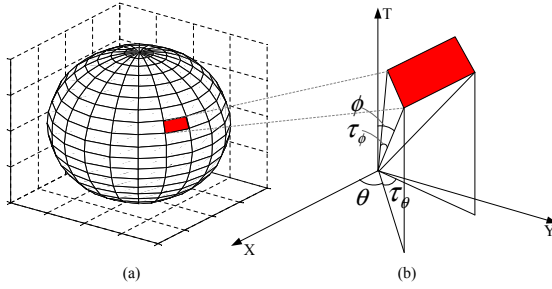
$$\{(x, y, t) | \rho = F(x, y, t; \phi, \theta), (x, y, t) \in \chi^3\} \quad (4)$$

where  $F(x, y, t; \phi, \theta)$  is the plane's representation in the spherical coordinates as in Equ.3 and  $\chi^3$  denotes the range of the definition of the coordinate  $(x, y, t)$ .

Theoretically, there is a one-to-one mapping between the planes in the STV space and the points in the Hough space with  $\phi$ ,  $\theta$  and  $\rho$  as axes. Taking Fig.2 for example, a plane  $S$  in the STV space corresponds to a point  $P(\phi_0, \theta_0, \rho_0)$  in the Hough space.

But as mentioned in previous section, for many applications in image processing and computer vision, we can seldom find a plane that strictly meets the geometry definition as in Equ.4 due to the imperfections in either the image data or the edge detector. In addition, due to the translation and rotation uncertainty, a nonideal plane in the STV space evidently does not occupy a single point in the Hough space but a cluster of points instead, as mentioned in [22]. In order to make the definition to accommodate these nonideal planes, we extend the definition of the planes in the Hough space by extending a single point  $P(\phi_0, \theta_0, \rho_0)$  into a cuboid region  $R$  parameterized by the center position  $(\phi_0, \theta_0, \rho_0)$  and the cuboid size  $(2\tau_\phi, 2\tau_\theta, 2\tau_\rho)$ . This means that all the facets that fall into this cuboid region in the Hough space will still be considered as a *plane*. This *plane* is not a conventional plane that can fulfill the restriction in Equ.4, but it is a





**Fig. 3.** The 3D orientation partition based on sphere polar coordinates

plane that can tolerate certain degree of rotation and translation uncertainty. This motivates us to generalize the definition of the plane as:

$$\begin{aligned} \{ (x, y, t) | \rho = F(x, y, t; \phi, \theta), (x, y, t) \in \chi^3, \\ |\rho - \rho_0| \leq \tau_\rho, |\phi - \phi_0| \leq \tau_\phi, |\theta - \theta_0| \leq \tau_\theta \} \end{aligned} \tag{5}$$

We refer to this definition as a *generalized plane*. The geometrical explanation of this definition is that a generalized plane can be a combination of facets that fall into a cuboid region in the Hough space. This endows the generalized plane with robustness to the uncertainty of rotation and translation. Three important properties of the generalized plane are summarized here:

1. It can represent the nonideal planes which can be discontinuous and even with certain level of rotation and translation uncertainty.
2. The robustness to the uncertainty of rotation and translation can be controlled quantitatively by the parameters  $(\tau_\phi, \tau_\theta, \tau_\rho)$ . More specifically, the robustness to rotation can be controlled quantitatively by  $(\tau_\phi, \tau_\theta)$  and we refer to it as the *rotation uncertainty*; the robustness to translation can be controlled by  $\tau_\rho$  and we refer to it as the *translation uncertainty*.
3. When we restrict the window size to zero, i.e.  $\tau_\phi = 0, \tau_\theta = 0,$  and  $\tau_\rho = 0,$  then the *generalized plane* can degenerate into normal plane as defined in Equ.4. Therefore, the *generalized plane* can be considered as a superset of the plane.

The advantage of Equ.5 is that it can incorporate the uncertainty control into the plane’s definition explicitly. Therefore, we can produce planes with different description characteristics by varying the uncertainty parameters that are specified by  $(\tau_\phi, \tau_\theta, \tau_\rho)$ .

### 4 Orientation-Space Partition in the STV Space

According to the description in section 3, the generalized plane is defined in the Hough space and can be considered as a  $2\tau_\phi \times 2\tau_\theta \times 2\tau_\rho$  cuboid region that

centered at  $(\phi_0, \theta_0, \rho_0)$ . However, the description of this generalized plane is in the STV space. Therefore, we need to back-project the cuboid region from the Hough space into the STV space.

Intuitively, the back-projection of this cuboid region in the STV space is a sandglass-shaped region. We refer to this region in the STV space as a *partition* to distinguish it from the cuboid region in the Hough space. Based on this extension, we can find a one-to-one mapping between a cuboid region in the Hough space and a partition in the STV space. We achieve this goal by orientation partition and space partition.

#### 4.1 Orientation Partition

Orientation partition is the back-projection of the angular uncertainty  $(\tau_\phi, \tau_\theta)$  from the Hough space to the STV space. As we have mentioned in previous section, the normal direction of the plane is determined by the parameters  $\phi$  and  $\theta$ , and they have very specific meanings in the spherical coordinates: the inclination  $\phi$  is the angle between the zenith direction and the normal direction; the azimuth  $\theta$  is the angle between the reference direction on the chosen plane and the projection of the normal direction on the plane. The space expanded by  $\phi$  and  $\theta$  can be represented on a unit sphere, as shown in Fig. 3. There is a one to one mapping between the points on this sphere and the unit directional vectors. Therefore, the partition on this unit sphere corresponds to a partition on the orientation space. We apply the 2-dimensional quantization on the unit sphere by step size  $\tau_\phi$  and  $\tau_\theta$ , as shown in Fig. 3(a). Thus, the unit sphere is divided into a group of disjoint patches, and the directional vectors that map to the same patch are quantized to the same direction. By this means, the uncertainty parameters  $\tau_\phi$  and  $\tau_\theta$  can be mapped from the Hough space to the STV space.

More specifically, given a point  $(x, y, t)$  on the spatial-temporal volume  $V$ , the first-order derivatives (using filter  $[1, 0, -1]$ ) of the intensity along the three directions are represented as  $(V_x, V_y, V_t)$ . Then the normal direction of this point can be calculated as:

$$\begin{cases} n_x = V_x/s \\ n_y = V_y/s \\ n_t = V_t/s \end{cases} \quad (6)$$

where  $s = \sqrt{V_x^2 + V_y^2 + V_t^2}$  is the strength of the gradient. The orientation parameters  $\phi$  and  $\theta$  can be calculated as:

$$\begin{cases} \theta = \arctan n_y/n_x \\ \phi = \arctan \sqrt{(n_x^2 + n_y^2)/n_t^2} \end{cases} \quad (7)$$

By Equ. 3, we can calculate the distance parameter  $\rho$  also. Therefore, any point on the STV can be represented by a septet  $[x, y, t, s, \phi, \theta, \rho]$ . Then we quantize the angles  $\phi$  and  $\theta$  by step size  $\tau_\phi$  and  $\tau_\theta$  respectively, according to the rotation

uncertainty defined in Equ.5. Thus far, the original STV has been divided into  $m \times n$  disjoint directional channels:

$$\{[x, y, t, s, \rho]_{\phi_1 - \theta_1}, \dots, [x, y, t, s, \rho]_{\phi_i - \theta_j}, \dots, [x, y, t, s, \rho]_{\phi_m - \theta_n}\} \quad (8)$$

where:

$m, n$  — number of index by quantization,  $m = \lceil \pi / \tau_\phi \rceil$ ,  $n = \lceil \pi / \tau_\theta \rceil$ ;  
 $\phi_i, \theta_j$  — the quantized inclination and azimuth angles,  $\phi_i = i * \tau_\phi$ ,  $\theta_j = j * \tau_\theta$ ;  
 $\phi_i - \theta_j$  — the symbol that represents the principal orientation of each channel;  
 For each channel  $[x, y, t, s, \rho]_{\phi_i - \theta_j}$ , only the voxels whose normal angle can be quantized as its principal orientation  $\phi_i - \theta_j$  are preserved and all the other voxels are set to zero. We refer to this operation as the orientation partition, as shown in Fig.4(a)-(c).

For simplicity, we will use  $V_{\phi_i - \theta_j}$  to denote the channel  $[x, y, t, s, \rho]_{\phi_i - \theta_j}$ . Therefore, the results of orientation partition can be represented as:

$$\{V_{\phi_i - \theta_j} | i = 1, \dots, m; j = 1, \dots, n\} \quad (9)$$

where  $\bigcup_{i=1, j=1}^{i=m, j=n} V_{\phi_i - \theta_j} = V$  and  $V_{\phi_i - \theta_j} \cap V_{\phi_p - \theta_q} = \emptyset, i \neq p, j \neq q$ .

## 4.2 Space Partition

Space partition is used to back-project the translation uncertainty  $\tau_\rho$  into the STV space. For each channel  $[x, y, t, s, \rho]_{\phi_i - \theta_j}$ , it can be further partitioned by a group of parallel planes, and we refer to these planes as the partition planes. Moreover, the normal directions of all the partition planes are equal to the principal direction  $\phi_i - \theta_j$  of the current channel and the distances between the adjacent partition planes are equal to the translation uncertainty parameter  $\tau_\rho$ . The region between the two adjacent partition planes can be considered as a generalized plane and all the voxels located within this region belong to the same generalized plane. By this means, we can explicitly control the plane's robustness to the translation uncertainty.

The space partitions can be now represented as  $[x, y, t, s]_{\phi_i - \theta_j}^{\rho_k}$ . We denote it by  $P_{\phi_i - \theta_j}^{\rho_k}$  for simplicity. These partitions fulfill the following definition:

$$\left\{ P_{\phi_i - \theta_j}^{\rho_k} | k = 1, \dots, o; \bigcup_{k=1}^o P_{\phi_i - \theta_j}^{\rho_k} = V_{\phi_i - \theta_j}; P_{\phi_i - \theta_j}^{\rho_m} \cap P_{\phi_i - \theta_j}^{\rho_n} = \emptyset, m \neq n \right\} \quad (10)$$

where  $o$  is the number of spatial partition, as shown in Fig.4(d).

Moreover, the strength of the gradient within each partition can be represented as:

$$g_{\phi_i - \theta_j}^{\rho_k} = q(P_{\phi_i - \theta_j}^{\rho_k}) \quad (11)$$

where  $q(\cdot)$  is the function that calculates the summation of gradient strength within a partition.

By the orientation and space partition, we can associate a cuboid region in the Hough space with a partition in the STV space. The representation property of the generalized plane can be controlled by  $(\tau_\phi, \tau_\theta, \tau_\rho)$  during the partition procedure. The overall orientation-space partition procedure can be seen in Fig.4. The statistical description of the generalized plane can be calculated easily within each partition, which will be detailed in the following section.

### 5 Computation of STGGP Descriptor

Thus far, we have mapped the generalized plane from the Hough space to the STV space by orientation-space partition. In this section, we will present how to calculate the descriptors for the generalized plane in the STV space.

The descriptor of the generalized plane is 9-dimensional heterogeneous vector. It can encode the gradient strength, position and shape information of the plane. For any channel  $V_{\phi_i - \theta_j}$ , its descriptor can be represented as:

$$(i_{max}, g_{max}, \sigma, m_x, m_y, m_t, v_{norm}, v_{tangX}, v_{tangY})_{\phi_i - \theta_j}.$$

Given a feature that is specified by a cuboid  $C(x_0, y_0, t_0, w, h, l)$  and the uncertainty parameters  $(\tau_\phi, \tau_\theta, \tau_\rho)$ , we firstly perform the orientation-space partition within  $C$  as mentioned above. Then for each channel  $V_{\phi_i - \theta_j}$ , we can get the space partitions  $P_{\phi_i - \theta_j}^{\rho_k}$  as Equ.10. The gradient strength  $g_{\phi_i - \theta_j}^{\rho_k}$  of each partition can be calculated as in Equ.11. In the following section, we will drop the orientation subscript  $\phi_i - \theta_j$  for simplicity. The items of the descriptor can be calculated as follows:

- $i_{max}$ : is the normalized index value of the space partition with the maximum gradient strength. It can be calculated as  $i_{max} = \frac{i'_{max}}{o}$ , where  $i'_{max} = \arg \max_k (g^{\rho_k})$  is the index of the space partition with the maximum gradient strength and  $o$  is the number of space portions.
- $g_{max}$ : is the normalized maximum gradient strength, where  $g_{max} = \frac{g'_{max}}{\sum_{k=1}^o g^{\rho_k}}$ ,  $g'_{max} = \max(g^{\rho_k})$ .
- $\sigma$ : is the standard deviation of the gradient strength, and can be calculated as  $\sigma = \sqrt{\frac{1}{o} \sum_{k=1}^o (g^{\rho_k} - \bar{g})^2}$ , where  $\bar{g} = \frac{1}{o} \sum_{k=1}^o g^{\rho_k}$ .
- $(m_x, m_y, m_t)$ : is the normalized mean position of all the non-zero points in partition  $P^{\rho'_{max}}$ , for example,  $m_x$  is calculated as follows:

$$m_x = \frac{1}{z} \sum_{i=0}^z \frac{(x_i - x_0)}{w} \tag{12}$$

where:

$z$  — the number of non-zero points in the partition  $P^{\rho'_{max}}$ ;

$x_0$  — the center point of the feature cuboid  $C$ ;

$w$  — the size of the feature cuboid  $C$ ;

- $(v_{norm}, v_{tangX}, v_{tangY})$ : denote the standard deviation of the non-zero points in partition  $P^{\rho'_{max}}$ , for example,  $v_{norm}$  is be calculated as follows:

$$v_{norm} = \sqrt{\frac{1}{z} \sum_{i=1}^z (r_{i,norm} - m_{norm})^2} \tag{13}$$

where  $r_{i,norm}$  and  $m_{norm}$  are the new position of the points and their means in the rotated coordinates;

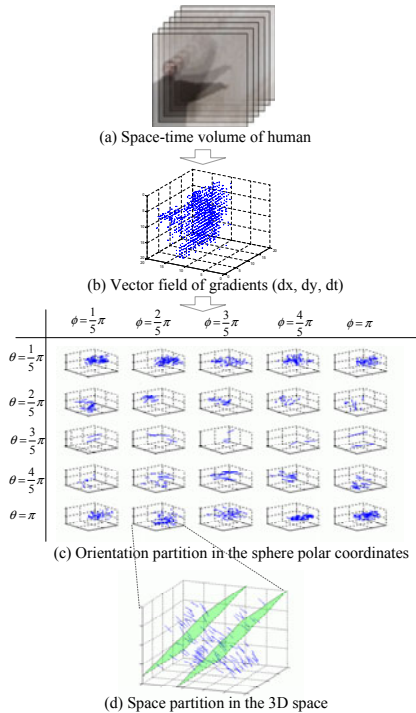
The reason for coordinate rotation is to align the normal direction of the generalized plane with the axis of the new coordinate frame. In this new coordinate

frame, the distribution of the non-zero points can be easily described. And the rotation matrix  $A$  of the current channel  $\phi_i - \theta_j$  is defined as:

$$\begin{aligned}
 A &= A_{\phi_i} A_{\theta_j} \\
 &= \begin{bmatrix} \cos \phi_i \cos \theta_j - \cos \phi_i \sin \theta_j - \sin \phi_i & & \\ \sin \theta_j & \cos \theta_j & 0 \\ \sin \phi_i \cos \theta_j - \sin \phi_i \sin \theta_j & \cos \phi_i & \end{bmatrix} \tag{14}
 \end{aligned}$$

The new coordinate frame is referred to as  $norm - tangX - tangY$ . In this new coordinate frame, the generalized plane is parallel to the  $tangX - tangY$  plane and its norm is aligned with the  $norm$  axis. In this new coordinate, the shape property of the generalized plane can be easily characterized. For example,  $v_{norm}$  is the standard deviation along the normal direction, and it can be considered as the "thickness" of the generalized plane; and  $(v_{tangX}, v_{tangY})$  can be used to describe the "shape" of the plane.

Thus far, for each channel  $V_{\phi_i - \theta_j}$ , we have obtained a 9-dimensional feature vector, i.e.,  $(i_{max}, g_{max}, \sigma, m_x, m_y, m_t, v_{norm}, v_{tangX}, v_{tangY})_{\phi_i - \theta_j}$ . Then, by concatenating the feature vectors of all the channels/orientations, we can get the final STGGP descriptor.



**Fig. 4.** The overview of the orientation-space partition on the spatial-temporal volume

## 6 Experiments

In this section, the proposed STGGP is evaluated on the public dataset. Firstly, since STGGP is a heterogeneous vector, we evaluate the contributions of the different components; secondly, we investigate how the temporal length affects the the performance of the method; thirdly, we evaluate the proposed method against the state of the art methods. In our experiments, the linear SVMs are used as the classifiers with parameter  $C = 0.1$ . For easy comparison, we plot the "recall" vs. "false positive per image" curve.

We use the ETHZ as the benchmarking dataset [29]. The size of normalized STV is  $96 \times 64 \times 5$  and the size of SGTPP feature is  $16 \times 16 \times 5$ . With 2 translation uncertainty settings,  $\tau \in \{4, 8\}$ , we use 78 STGGP features for representing a STV. The orientation uncertainty parameters are set as:  $\tau_\phi = \pi/5$  and  $\tau_\theta = \pi/5$ , therefore, there are 25 orientation channels. The overall dimension of a STV descriptor is 17550.

Since STGGP is a heterogeneous feature vector, it contains the gradients' strength, position and shape information. Therefore, in the first experiment, it is worth to evaluate the contributions of these different components. We reorganized the components of the GGP descriptors as follows:

- *STGGP\_C1*: ( $g_{max}$ ) only contains the maximum gradients' strength of the partitions, and its description ability is close to HOG.
- *STGGP\_C2*: ( $g_{max}, i_{max}, \sigma$ ) adds partition index and the standard deviation of the gradient strength to represent the strength distribution information within the feature region.
- *STGGP\_C3*: ( $g_{max}, i_{max}, \sigma, m_x, m_y$ ) adds the mean positions of all the non-zero pixels to represent the position information.
- *STGGP*: ( $g_{max}, i_{max}, \sigma, m_x, m_y, v_{norm}, v_{tang}$ ) the STGGP descriptor that adds the standard deviations of positions to represent the shape of non-zero pixels in the partition.

The evaluation results can be seen in Fig 5(a). From these results, we can make a few observations: firstly, the performance can be improved monotonically as long as the new components are heterogeneous to the previous ones; secondly, the position information is critical of the performance, and the most prominent improvements can be observed after the position information have been added.

In the second experiment, we evaluate how the number of frames can affect the performance of STGGP. Therefore, we re-generate the sample sets with different frame numbers and train the detectors from these sets. Here we use *STGGP\_fn* to represent the detectors that trained with different frame numbers and  $n$  is the number of frame. For *STGGP\_f1*, we just use the original GGP detector that use only the static features. The results of these detectors can be seen in Fig 5(b). When we add frame number from 1 to 5, a monotonously improvement can be observed; but when we add more frames, the performance actually dropped. A possible explanation of is that for a longer temporal duration, the ego-motion of the camera will substantially affect the spatial-temporal shape of the human.

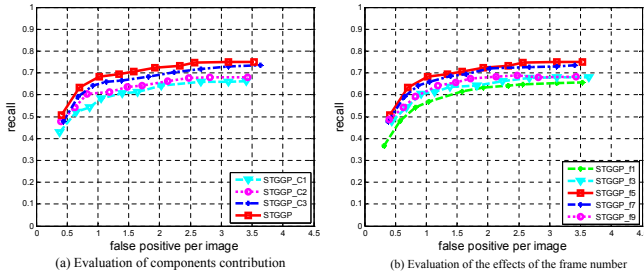


Fig. 5. Parameter evaluation on ETH-01 set

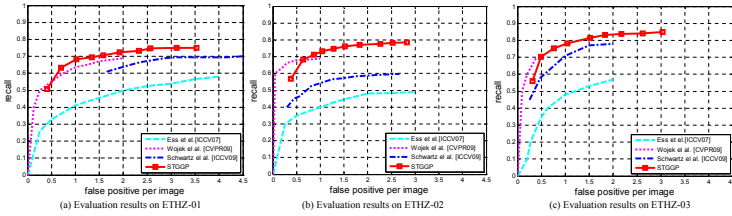


Fig. 6. Evaluation results on the ETHZ human dataset

In the third experiment, the STGGP detector is evaluated against the state of the art methods [29][25][3]. On ETH-01 set, the STGGP yields comparable results to the best results in [25], as shown in Fig 6(a); On ETH-02 and ETH-03 sets, the STGGP outperforms the other methods, as shown in Fig 6(b)(c). Another observation is that the features combining both appearance and motion outperform the appearance only based detector by a big margin. Some samples of the detection results can be found in Fig 7

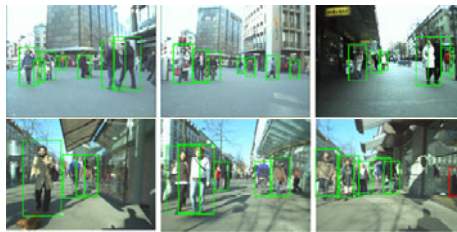


Fig. 7. Sample detection results on ETHZ dataset

## 7 Conclusion

In this paper we have developed a spatial-temporal granularity-tunable gradients partition (STGGP) descriptor to represent the human’s motion pattern

in the spatial-temporal domain. Firstly, the *generalized plane* is defined in the Hough space. By incorporating the rotation and translation uncertainties in the definition of the plane, it can describe the object with a family of descriptors with different representation ability, from the detailed geometrical representation to the statistical description. Then, by orientation-space partition, the generalized plane can be back-projected from the Hough space to the spatial-temporal space. Finally, we form the heterogeneous descriptor in the generalized plane. The heterogeneous descriptor contains gradient's strength and spatial distribution information, which further improve its representation ability. The STGGP descriptor is tested for human detection in image sequences and promising results have been achieved.

## Acknowledgement

This paper is partially supported by Natural Science Foundation of China under contracts No.60772071, No.60832004, No.60872124, and No. U0835005; National Basic Research Program of China (973 Program) under contract 2009CB320902. The financial support from the Infotech Oulu is also gratefully acknowledged.

## References

1. Han, F., Shan, Y., Sawhney, H.S., Kumar, R.: Discovering class specific composite features through discriminative sampling with swendsen-wang cut. In: CVPR (2008)
2. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: ICCV, pp. 32–39 (2009)
3. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. In: ICCV (2009)
4. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)
5. Dollar, P., Babenko, B., Belongie, S., Perona, P., Zhuowen, T.: Multiple component learning for object detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 211–224. Springer, Heidelberg (2008)
6. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR (2009)
7. Ott, P., Everingham, M.: Implicit color segmentation features for pedestrian and object detection. In: ICCV, pp. 724–730 (2009)
8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, pp. 511–518 (2001)
9. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV 38, 15–33 (2000)
10. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. TPAMI 23, 349–361 (2001)
11. Gavrila, D.M.: Pedestrian detection from a moving vehicle. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 37–49. Springer, Heidelberg (2000)
12. Lin, Z., Davis, L.S., Doermann, D., DeMenthon, D.: Hierarchical part-template matching for human detection and segmentation. In: ICCV (2007)



13. Lin, Z., Davis, L.S.: A pose-invariant descriptor for human detection and segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 423–436. Springer, Heidelberg (2008)
14. Ferrari, V., Tuytelaars, T., Gool, L.V.: Object detection by contour segment networks. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 14–28. Springer, Heidelberg (2006)
15. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: ICCV (2005)
16. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
17. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 69–82. Springer, Heidelberg (2004)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
19. Zhu, Q., Avidan, S., Yeh, M.C., Cheng, K.T.: Fast human detection using a cascade of histograms of oriented gradients. In: CVPR, pp. 1491–1498 (2006)
20. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: CVPR (2008)
21. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: CVPR (2007)
22. Liu, Y., Shan, S., Zhang, W., Gao, W., Chen, X.: Granularity-tunable gradients partition (ggp) descriptors for human detection. In: CVPR, pp. 1255–1262 (2009)
23. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
24. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: ICCV, pp. 734–741 (2003)
25. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: CVPR (2009)
26. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. In: CVPR, vol. 2, pp. 123–130 (2001)
27. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: CVPR, vol. 2, pp. 1395–1402 (2005)
28. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: ICCV, vol. 1, p. 166–173 (2005)
29. Ess, A., Leibe, B., Gool, L.V.: Depth and appearance for mobile scene analysis. In: ICCV, pp. 14–21 (2007)

# Being John Malkovich

Ira Kemelmacher-Shlizerman<sup>1</sup>, Aditya Sankar<sup>1</sup>,  
Eli Shechtman<sup>2</sup>, and Steven M. Seitz<sup>1,3</sup>

<sup>1</sup> University of Washington  
{kemelmi,aditya,seitz}@cs.washington.edu

<sup>2</sup> Adobe Systems  
elische@adobe.com

<sup>3</sup> Google Inc.

**Abstract.** Given a photo of person A, we seek a photo of person B with similar pose and expression. Solving this problem enables a form of *puppetry*, in which one person appears to control the face of another. When deployed on a webcam-equipped computer, our approach enables a user to control another person’s face in real-time. This image-retrieval-inspired approach employs a fully-automated pipeline of face analysis techniques, and is extremely general—we can puppet anyone directly from their photo collection or videos in which they appear. We show several examples using images and videos of celebrities from the Internet.

**Keywords:** Image retrieval, facial expression analysis, puppetry.

## 1 Introduction

“Ever wanted to be someone else? Now you can.”  
—tagline from the film *Being John Malkovich*

In the film *Being John Malkovich*, a puppeteer (played by John Cusack) discovers a portal that allows him to control the real life movements of John Malkovich (played by himself). While puppeteering real people might seem a bit far fetched, it should be possible to control *digital likenesses* of real people. In particular, we seek to construct a photographic simulation (i.e., avatar) of John Malkovich that you can control by moving your face; when you smile, move your head, or close your eyes, you see John Malkovich doing the same.

One way to attack this puppetry problem might be to create a photo-realistic 3D model of John Malkovich’s head, instrumented with several degrees of freedom (e.g., mouth open, head rotate, etc.), and map the user’s head motions to the model. Indeed, most prior work on avatars and puppetry has followed a similar approach [1,2,3,4]. However, creating a sufficiently accurate model of a real person is a major challenge, particularly if we don’t have access to the actor to pose for a scanning session.

Instead, we recast the puppetry problem as image retrieval: given a query image or video of person A (the user), and a set of images of person B (John



**Fig. 1.** One person’s expressions (top row) are mapped to another person’s face (bottom row) by real-time matching to an image database. In this case, the input is a video of Cameron Diaz, and the database is formed from a video (John Malkovich, bottom-left 4 images) or an unstructured set of photographs downloaded from the Internet (George W. Bush, bottom-right 4 images).

Malkovich), find and display the best matching image or image sequence of person B. This approach has a number of key advantages, as follows. First, we avoid the complexity and technical difficulty of creating a 3D model and parameterizing expressions. Second, because the output are real photos, we can capture all the complexities of the face (hair, light scattering, glasses, etc.) that are difficult to simulate. And finally, the approach operates on just about any set of photos or video, and is fully automatic. I.e., it is possible to create an avatar simply by typing an actor’s name on an image/video search site and processing the resulting images and/or videos. The approach can also be used to drive one video with another; Fig. 1 shows Cameron Diaz driving John Malkovich and George W. Bush.

The main challenge to making this image retrieval approach work is defining a metric that can reliably match an image of person A to an image of person B with similar pose and expression. Significantly complicating this task is the fact that the facial characteristics, lighting, and sex of the two people may be different, resulting in large appearance variation between person A and person B. The main contribution of this paper, in addition to posing puppetry as image retrieval, is a processing pipeline that yields high-quality real-time facial image retrieval, and that operates reliably on both video and unstructured photo collections. While this pipeline is based on existing pieces from the literature, we argue that it is not at all straightforward to create a real-time system that achieves the results presented here; the contribution is the system and the novel application.

Our approach operates as follows: images of the target face (person B) are processed using a combination of face detection, extracting fiducial features (e.g., eyes, nose, mouth), estimating pose, and aligning/warping to a frontal pose. The user (person A) is tracked to determine head-pose at video rates. After alignment, the user’s face is compared to all aligned images of the target face using a fast implementation of Local Binary Patterns (LBP) and chi-square

matching, and the best match is returned. We incorporate additional terms for temporal coherence. We've found the resulting approach to work remarkably well in practice. Fig. 2 shows our interactive web-cam-based system in action for a case of a user driving George Clooney. Video captures can be found on the paper's website: <http://grail.cs.washington.edu/malkovich/>.

## 1.1 Related Work

There is quite a large literature on avatars, puppetry, and performance-driven animation. We therefore limit our discussion to methods that specifically involve tracking video of a user's face to drive the appearance of a different person or model. And while tracking mocap markers to drive scanned or hand-crafted animated models is a mainstay of digital special effects (famous examples in movies include *Polar Express* and *Avatar*), we focus here on markerless solutions that operate from photos alone.

While there are a number of markerless puppetry techniques, the vast majority of these methods assume the availability of a 3D model of the target face, e.g., [3]. A very recent example is the work of Weise et al. [4], who demonstrate very impressive puppetry results using a real-time structured light scanner to drive a previously captured 3D model. 3D puppetry via face tracking is starting to become mainstream—Logitech's webcams now come with software that tracks a user's face and gestures to control an animated on-screen avatar.

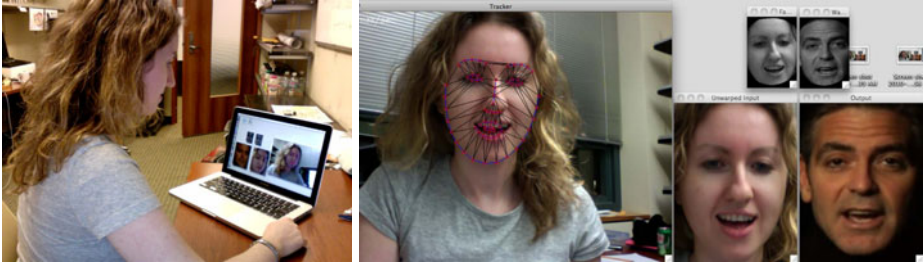
Pighin et al. [1] were among the first to demonstrate purely image-based face capture and puppetry. In this work, the model was created by manually specifying a set of correspondences between features on a 3D head model and features in several photos of the person. More recent work in this vein includes Zhang et al. [2] who used video to create the 3D model and simplified the manual work to 5 features in two views.

Although they do not relate to puppetry per se, we are inspired by Kumar et al.'s face search [5] and Bitouk et al.'s swapping [6] work, which operate robustly on large collections of images downloaded from the Internet, and Goldman et al. [7] who enable mouse-based face posing from a database of tracked video frames.

We note however, that no prior work has enabled puppetry with arbitrary, unstructured photo collections. This capability dramatically broadens the applicability of puppetry techniques, to *any* person whose photos are available.

## 2 Overview

Our system allows fully automatic real time search of similar facial expressions of a target person given image queries from a webcam. The user can be any person; there is no need to train the system for a specific user. The user can make expressions to the camera, as well as change the pose of the head, and get in real time similar expressions and poses of the target face. The target face can be represented by a video or by a set of photos (e.g., photos of a celebrity downloaded from the Internet). In each query frame the face is tracked



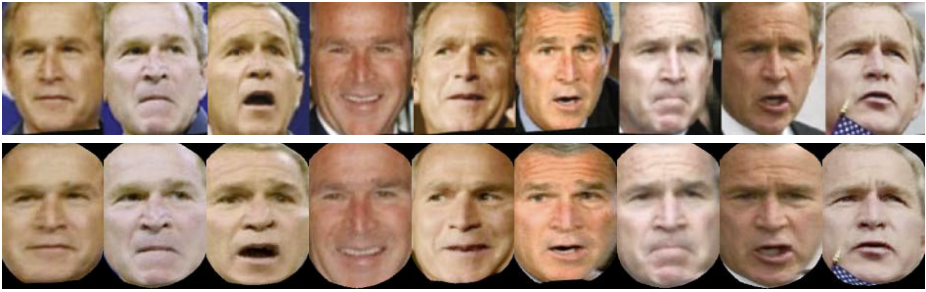
**Fig. 2.** Our puppeteering system. Left - our setup. Right - screen capture of our system: the face tracker applied on an input webcam video; Top grayscale small pair - the input face after cropping and warping to frontal view and the matching target face found by our method; Bottom large pair - user face and matching target image (raw input).

automatically, the 3D position of the face is recovered, then the detected face region is fitted to a template 3D model of a face and is warped to a frontal pose. In the process, we estimate the location of the eyes and mouth of the user’s face. We consider each of these regions independently and for each region compute a Local Binary Patterns (LBP) feature vector. The same is done for each photo (or video frame in case a movie is available) of the target person. We then compute distances between the mouth region of the target face and the mouth region of the user face, and similarly for the eyes regions. These two distances define our appearance distance. In addition, we compute the distance between the 3D pose of the user and the 3D pose of the target in each image, and a temporal continuity distance. These three distances are combined together to find the best match in terms of appearance, pose and continuity. We describe our geometric alignment method in Section 3, and the appearance representation in Section 4. In Section 5 we present our distance function. Results and evaluations of the method are presented in Section 6.

### 3 Image Alignment to Canonical Pose

In this section we present a framework to align the images of the user and target faces to a canonical (frontal) pose. The input to the method is a live video feed (e.g., webcam) or a video of a person. We first automatically track the face in each frame of the video, using the algorithm of Saragih et al. [8]. The aim here is to estimate the location of the face and its pose in each given frame, and use these to perform warping of each image to a canonical pose.

[8] is based on fitting a parametrized shape model to an image such that its landmarks correspond to consistent locations on the face. In particular, in each frame, predictions regarding locations of the model’s landmarks are made by utilizing an ensemble of local feature detectors, and then combined by enforcing a prior over their joint motion. The distribution of the landmark locations is represented non-parametrically and optimized via subspace constrained mean-shifts.



**Fig. 3.** Results of our warping procedure (based on 3D pose estimation and using 3D template model). Top row: images aligned in 2D. Bottom row: warped images (3D alignment).

For the target person, in case we have a video available we estimate the location of the face and landmarks using the same procedure as the user’s face. In case the target face is represented by a collection of photos we cannot use tracking. We instead apply a face detector [9] followed by a fiducial points detector [10] that provides the landmarks (the left and right corners of each eye, the two nostrils, the tip of the nose, and the left and right corners of the mouth). Given the landmark positions we recover the 3D position of the face. For this we use a neutral face model from the publicly available spacetime faces dataset [11] as our template model. Given the points on the image and the corresponding pre-labeled points on the 3D template model we first subtract the centroid from each of the point arrays, recover a linear transformation between them and then find rotation and scale using RQ decomposition. The yaw, pitch and roll angles are then estimated from the rotation matrix.

Given the estimated pose we can transform the template model to the orientation of the face in the image, and consequently warp the image to a frontal pose. In Figure 3 we show a few examples of images warped using this procedure.

## 4 Appearance Representation

Once the images have been aligned and warped to a frontal pose, the next step is to compare the appearance of the faces to find similar facial expressions. Since all images are aligned to a template model we can identify the areas in the image that correspond to different face regions. In this paper we concentrate on the regions that correspond to eyes and mouth, however one can consider comparing other regions as well (e.g., position of eyebrows).

To compare appearance of facial regions we have chosen to use the Local Binary Pattern (LBP) histograms [12], which have previously proven effectiveness for face recognition [13] and facial expression recognition. These methods however were applied on frontal faces captured with similar conditions (lighting, resolution etc.). In our case (especially in the case of unstructured photo collections) these conditions do not often hold. However our alignment step that

warps the face to frontal position compensates for pose differences and allows us to effectively use LBP features for comparison of facial regions.

LBP operates by converting each pixel into a code which encodes the relative brightness patterns in a neighborhood around that pixel. In particular, each neighbor is assigned a 1 or 0 if it is brighter or darker than the center pixel. This pattern of 1's and 0's defines a binary code that is represented as an integer. Explicitly the LBP code is defined as:

$$LBP(c) = \sum_{p=0}^{|\mathcal{N}|-1} 2^p H(I_p - I_c), \quad (1)$$

where  $H(x) = 1$  if  $x > 0$  and 0 otherwise,  $I_c$  and  $I_p$  are the intensities of the center pixel and neighboring pixel correspondingly, and  $\mathcal{N}$  is a set of neighbors of the center pixel  $c$ . The histogram of these codes defines the descriptor for each facial region. For example in case the neighborhood around a pixel is chosen to be 3x3 square, there are 8 neighbors, and so there are  $2^8 = 256$  labels (or bins in the histogram). Intuitively each code can be considered as a micro pattern, that encodes local edges, homogenous areas and other primitives. The binarization quantization achieves robustness to small lighting changes and robustness to small motions is obtained by forming the histogram. Following [12] for each pixel we use a circular neighborhood around it and bilinearly interpolate values at non-integer pixel coordinates. We further use the extended version of the operator, called uniform code, that reduces the length of the feature vector. A code is called uniform if it contains at most two transitions between 0 to 1 or vice versa. In the computation of the LBP histogram each uniform code has its own label and all non-uniform codes get a single label.

## 5 Distance Metric

Distance between two facial regions is defined by  $\chi^2$ -distance between the corresponding LBP descriptors.  $\chi^2$  is defined as:

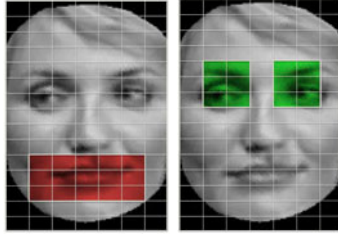
$$\chi^2(x, y) = 1/2 \sum_i (x_i - y_i)^2 / (x_i + y_i), \quad (2)$$

where in our case  $x$  and  $y$  are two LBP descriptors. To compare the mouth region we divide it to 3x5 cells and sum up the distances of all cells, each eyes region is divided to 3x2 cells (the whole face is divided to 15x7 cells). Figure 4 shows the masks we use.

Given an input image  $i$  we compare it to all target's images  $j$ . Our appearance distance function between each frame  $i$  and image  $j$  is defined as

$$d_{appear}(i, j) = \alpha^m d^m(i, j) + \alpha^e d^e(i, j) \quad (3)$$

where  $d^{\{m,e\}}$  are the LBP histogram  $\chi^2$ -distances restricted to the mouth and eyes regions, respectively, and  $\alpha^{\{m,e\}}$  are the corresponding weights for these



**Fig. 4.** The regions we use in our appearance distance. The image is aligned to a canonical pose and divided to 15x7 cells. The mouth region (marked in red) is divided to 3x5 cells and each eye region (marked in green) is divided to 3x2 cells.

regions. For example, assigning  $\alpha^m = 1$  and  $\alpha^e = 0$  will result in only the mouth region being considered in the comparison. Prior to the combination of the mouth and eyes distances we normalize each of these by subtracting the minimum distance (over the target images) and dividing by the maximum distance.

Our complete distance function also includes difference in pose and a temporal continuity term. The difference in pose is measured separately for yaw  $Y$ , pitch  $P$  and roll  $R$ , and each of these is normalized using a robust logistic function. The pose term is:

$$d_{pose}(i, j) = L(|Y_i - Y_j|) + L(|P_i - P_j|) + L(|R_i - R_j|) \quad (4)$$

where the logistic function  $L(d)$  is defined as

$$L(d) = \frac{1}{1 + e^{-\gamma(d-T)/\sigma}} \quad (5)$$

with  $\gamma = \ln(99)$ . It normalizes the distances  $d$  to the range  $[0, 1]$ , such that the value  $d = T$  is mapped to 0.5 and the values  $d = T \pm \sigma$  map to 0.99 and 0.01 respectively. The temporal continuity term computes the appearance distance between the previous input frame  $i - 1$  and all the target images  $j$ . The complete distance function is then:

$$D(i, j) = d_{appear}(i, j) + \alpha^p d_{pose}(i, j) + \alpha^t d_{appear}(i - 1, j) \quad (6)$$

where  $\alpha^p, \alpha^t$  are the weights of the pose and continuity terms. The best match per input frame is the target image that minimizes  $D(i, j)$ .

## 6 Results

In this section we give details on our experimental setup. The performance of the method is much better conveyed through videos; for the video captures and more results please see the paper's website: <http://grail.cs.washington.edu/malkovich/>. We have experimented with controlled experiments as well as videos of celebrities





(a) Full measure



(b) Without mouth similarity



(c) Without eyes similarity

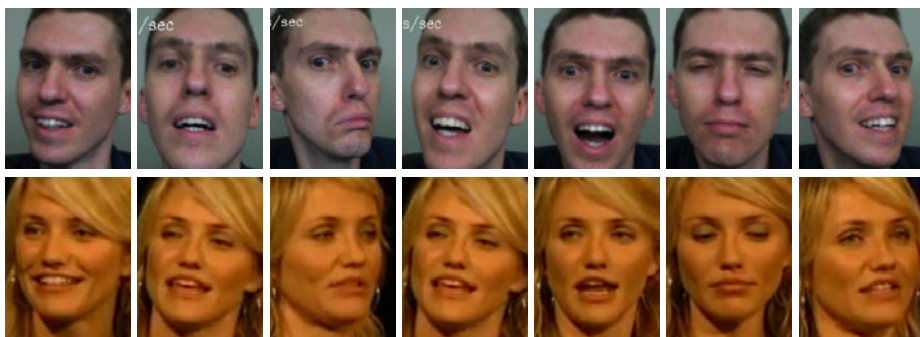
**Fig. 5.** Puppeteering evaluation. We recorded a video of person A (70 frames) and a video of person B of similar length. (a) 7 frames from person A video (first row); The corresponding frames of person B using the combined measure - mouth+eyes+pose (second row); (b) The corresponding frames *without mouth* measure - only expressions with high correlation between the eyes and mouth (like surprise) have similar mouth expression (third row). (c) Person A and the corresponding matches of B *without eyes* measure - the eyes are flickering across consecutive output frames.

downloaded from the Internet<sup>1</sup>. We also used the unstructured collection of photos of George W. Bush from the LFW database [14] as a target dataset. We begin by describing the implementation details of our system and then describe the experimental results.

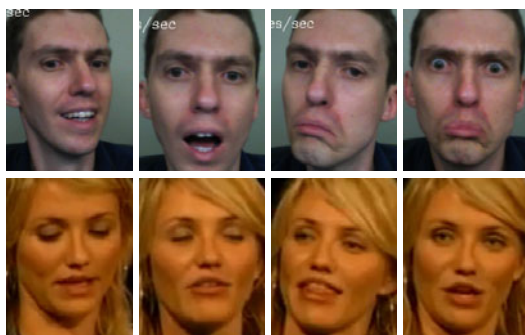
<sup>1</sup> Cameron Diaz - <http://www.youtube.com/watch?v=fWHgZz809Pw>  
 George Clooney - <http://www.youtube.com/watch?v=iZyw5-Sm0Zk>  
 John Malkovich - <http://www.mefeedia.com/watch/23930904>



**Fig. 6.** Puppeteering George Clooney. A few frames of the user captured by a webcam, followed by the corresponding retrieved faces of George Clooney (the target database consists of 1197 video frames).



(a) A sample of good matches



(b) Some failure cases

**Fig. 7.** Puppeteering Cameron Diaz. (a) A few frames of the user captured by a webcam, followed by the corresponding retrieved faces of Cameron Diaz (the database is a video of 1240 frames). (b) Some failure cases - most failures are due to a combination of an expression with pose of the user that do not exist in the target database. In this example the proportion of good/bad matches was around 0.7/0.3.



**Fig. 8.** Puppeteering an unstructured dataset of George W. Bush. A few frames of the user captured by a webcam, followed by the corresponding retrieved faces of George W. Bush (the target database is a collection of 870 photographs of George W. Bush).

### 6.1 Implementation Details

We use the following parameters for all experiments:  $\alpha^m = \alpha^e = 1$ ,  $T = 5$ ,  $\sigma = 2$ ,  $\alpha^p = \alpha^y = \alpha^r = 0.2$ ,  $\alpha^t = 0.2$ . Before we apply our distance function we ignore from consideration target images that differ in pose from the user image by more than  $5^\circ$  (for yaw, pitch and roll). The LBP histogram is calculated per image cell using Gaussian weighting as a function of pixel’s distance from the center of the cell. The sigma we used is the width of the cell with a margin in the size of half of the cell.

The system runs at 7fps on a 2.26GHz Intel Core 2 Duo Macbook Pro. The images or video used to create the target dataset are processed using the same pipeline as the input video of the user, i.e., tracking the face (or detecting in case of unstructured photo collection), estimating the pose in each frame and calculating the feature vectors. When constructing the target dataset from a video, we sample every 3rd frame of the video. Processing a video of 1000 frames takes approximately 2.5 minutes.

### 6.2 Controlled Experiments

To evaluate performance, we captured videos of two people with different facial characteristics making similar facial expressions and used one person’s video to drive the other video. We also evaluated the effect of comparing different regions of the face (eyes only or mouth only) on overall performance. Figure 5 shows the results of this experiment. We can see that the match is remarkably good when both eyes and mouth are used, despite the different facial appearance of these two users (note that one is Asian and has a mustache while the other has neither of these characteristics). When the mouth is omitted in the metric, the pose and eyes are matched, but the expression remains relatively constant, except for the example in column 4, where eyes of the “surprise” expression are well-correlated with the mouth. Similarly, when the eyes are left out of the distance metric, the output sequence exhibits random blinks.



**Fig. 9.** Puppeteering John Malkovich with a video of Cameron Diaz

### 6.3 Videos of Celebrities

We downloaded and processed videos and still photos for several celebrities from the Internet. Figure 6 shows an example of puppeteering George Clooney; the user (top row) makes facial expressions and the best matching frame from George Clooney’s video is shown at the bottom. Note how both, the eyes of the user and the eyes of George Clooney close in the 3rd example from the left, and how the mouth changes are quite consistent. Figure 7 shows results of puppeteering Cameron Diaz, (a) shows a sample of the good matches and in (b) we show some failure cases. Most of the failure cases seem to be due to the combination of an expression and pose that do not exist in the video/collection of photos of the target face. Similarly, we show an example where a user is able to drive an unstructured photo collection of George W. Bush obtained from the Internet (Figure 8). We also show an example of driving a video using another video in Figure 9. More examples are shown in Figure 11.

In our experiments, we observed that when the user and the target face are of the same gender (woman to woman and man to man) the output sequence is smoother and better captures the expressions, due to similar facial features. However, we observed that the method also works quite well with a man driving a woman and vice versa (as shown in these figures).

## 7 Conclusions

We presented a real-time puppetry system in which the user can make a celebrity or other person mimic their own facial gestures. As with traditional puppetry, part of the fun is learning how to master the controls. In particular, the user often learns to best drive the celebrity (rather than the other way around); to make John Malkovich smile, the user may have to smile in a similar style to the celebrity.

Unlike most prior work in this area which maps an image to a model, our formulation is photo to photo, using metrics that seek to match facial pose and eye/mouth similarity. The key advantage of this approach is its generality—it operates fully automatically and works on just about any video or photo collection of a person.

Beyond our puppetry application, this is also a general solution for face image retrieval, i.e., one can search for photos by acting out a particular expression and pose. In addition this allows to use unlabeled datasets and to retrieve facial expressions that are difficult to define with keywords.

There are several aspects of performance that could be improved. While LBP provides some robustness to lighting changes, shadows and other strong effects sometimes bias the match to similar lighting instead of similar expression. Better tracking and modeling of head shape could also increase the operating range, particularly with near-profile views. Finally, we use a first order model for temporal coherence; a more sophisticated model could result in temporally smoother output.

**Acknowledgments.** The authors gratefully acknowledge Jason Saragih for providing the face tracking software [8]. This work was supported in part by Adobe and the University of Washington Animation Research Labs.

## References

1. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., Salesin, D.H.: Synthesizing realistic facial expressions from photographs. In: SIGGRAPH, pp. 75–84 (1998)
2. Zhang, Z., Liu, Z., Adler, D., Cohen, M.F., Hanson, E., Shan, Y.: Robust and rapid generation of animated faces from video images: A model-based modeling approach. *Int. J. Comput. Vision* 58, 93–119 (2004)
3. Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. *ACM Trans. Graph.* 24, 426–433 (2005)
4. Weise, T., Li, H., Gool, L.V., Pauly, M.: Face/off: Live facial puppetry. In: Symposium on Computer Animation (2009)
5. Kumar, N., Belhumeur, P., Nayar, S.: Facetracer: A search engine for large collections of images with faces. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 340–353. Springer, Heidelberg (2008)
6. Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., Nayar, S.K.: Face swapping: automatically replacing faces in photographs. In: SIGGRAPH, pp. 1–8 (2008)
7. Goldman, D.B., Gonterman, C., Curless, B., Salesin, D., Seitz, S.M.: Video object annotation, navigation, and composition. In: UIST, pp. 3–12 (2008)
8. Saragih, J.M., Lucey, S., Cohn, J.: Face alignment through subspace constrained mean-shifts. In: ICCV (2009)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, pp. 511–518 (2001)
10. Everingham, M., Sivic, J., Zisserman, A.: Hello! My name is. Buffy – automatic naming of characters in TV video. In: BMVC (2006)

11. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: high resolution capture for modeling and animation. In: SIGGRAPH, pp. 548–558 (2004)
12. Ojala, T., Pietikinen, M., Menp, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. PAMI* 24, 971–987 (2002)
13. Ahonen, T., Hadid, A., Pietikinen, M.: Face description with local binary patterns: Application to face recognition. *IEEE Trans. PAMI* 28, 2037–2041 (2006)
14. 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)

# Facial Contour Labeling via Congealing

Xiaoming Liu, Yan Tong, Frederick W. Wheeler, and Peter H. Tu

Visualization and Computer Vision Lab  
GE Global Research, Niskayuna, NY 12309  
{liux,tongyan,wheeler,tu}@research.ge.com

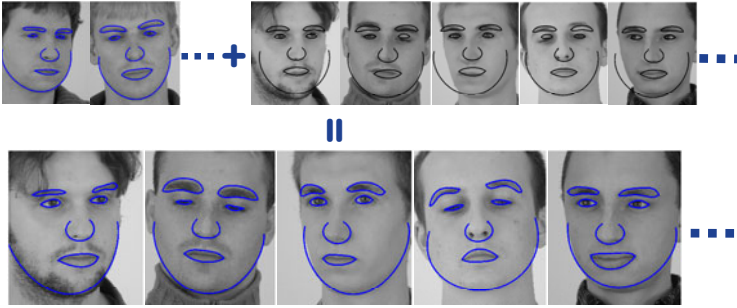
**Abstract.** It is a challenging vision problem to discover non-rigid shape deformation for an image ensemble belonging to a single object class, in an automatic or semi-supervised fashion. The conventional semi-supervised approach [1] uses a congealing-like process to propagate manual landmark labels from a few images to a large ensemble. Although effective on an inter-person database with a large population, there is potential for increased labeling accuracy. With the goal of providing highly accurate labels, in this paper we present a parametric curve representation for each of the seven major facial contours. The appearance information along the curve, named *curve descriptor*, is extracted and used for congealing. Furthermore, we demonstrate that advanced features such as Histogram of Oriented Gradient (HOG) can be utilized in the proposed congealing framework, which operates in a dual-curve congealing manner for the case of a closed contour. With extensive experiments on a 300-image ensemble that exhibits moderate variation in facial pose and shape, we show that substantial progress has been achieved in the labeling accuracy compared to the previous state-of-the-art approach.

**Keywords:** Facial contour, congealing, semi-supervised, ensemble, HOG.

## 1 Introduction

This paper addresses the problem of estimating semantically meaningful *facial contours* from an image ensemble using *semi-supervised congealing*. The shape of an object can be described by object contours, which include both the overall object boundary and boundaries between key components of the object. By facial contour, in particular, we refer to the boundary of chin and cheek, as well as the facial features including eyes, eyebrows, nose, and mouth. Given a large set of face images, semi-supervised congealing [1, 2] is defined as a process of propagating the labeling, which is the facial contour in this work, across the entire ensemble from a few labeled examples (See Fig. 1).

There are many applications of semi-supervised congealing. In computer vision, landmark labeling is necessary for learning models of the object shape, such as Active Appearance Models (AAM) [3, 4] and Boosted Appearance Model [5] for faces, which is often conducted manually for a large set of object instances/images. However, this is a labor-intensive, time-consuming, and error-prone process. Our semi-supervised approach will dramatically alleviate this problem.



**Fig. 1.** Given an image ensemble with an overlaid initial contour via face detection (top right), together with manual labels of contours on a few images (top left), our proposed algorithm estimates the contour parameters for all images in the ensemble (bottom), regardless of the variations of facial shape, pose, and unseen subjects.

Furthermore, our approach can be used to discover the non-rigid shape deformation of a real-world object, when applied to an image ensemble of an object class.

Given the wide application space of semi-supervised congealing, there is a surprisingly limited amount of prior work concerning ensemble-based non-rigid shape estimation for objects with greatly varying appearance, such as faces. The work by Tong *et al.* [1] might be the most relevant one to ours. They use least-square-based congealing to estimate the set of landmarks for all images in an ensemble given the labels on a few images. The least square term between any image pair is evaluated on a common rectangle region, which is where the image pair warps toward based on the landmark location. By gradually reducing the size of rectangle, the precision of landmark estimation is improved.

Although [1] has shown some promise, the accuracy of the labeling has potential for further improvement. First of all, the coarse-to-fine scheme and measurement in the warped space poses fundamental limitation on the accuracy. Also, the intensity feature is not salient enough to capture edge information, which is where all landmarks reside. To alleviate these problems, we propose a novel approach in this paper. With the goal of providing high accuracy in labeling, we use a parametric curve to represent the facial contour, rather than a landmark set. Hence, the appearance feature along the curve, named *curve descriptor*, can be extracted and drives the congealing process. Since two curve functions are used to represent a closed contour such as the eye, we present a dual-congealing algorithm operating jointly on both curves, with the help of a geometric constraint term. We demonstrate that advanced features such as HOG [6] can be utilized in the proposed congealing framework. With extensive experiments, we show that large progress has been achieved in the labeling accuracy compared to the state-of-the-art approach.



## 2 Prior Work

There is a long history of unsupervised group-wise registration in computer vision [7], particularly in the area of medical image analysis. Learned-Miller [2, 8] names this process “congealing”, where the basic idea is to minimize a cost function by estimating the warping parameters of an ensemble. The work by Cox *et al.* [9] is a recent advance in least-squares-based congealing (LSC) algorithm. However, these approaches estimate only affine warping parameters for each image, rather than the non-rigid deformation addressed here.

There is also work on unsupervised image alignment that allows more general deformation models, such as [10–18]. However, almost all approaches report results on images with small intra-class appearance variation, such as brain image, digits, and faces of a small population. In contrast, the semi-supervised congealing algorithm of [1] demonstrates promising performance on an ensemble of over 1000 images from hundreds of subjects, which motivates us to use the semi-supervised approach for facial contours.

There is a rich literature concerning contour and edge detection [19]. We should note that in dealing with real-world images, the dominant edge from low-level image observations might not be consistent with the high-level semantic-meaningful contour. For example, the double eyelid can have stronger edge information compared to the inner boundary between the conjunctiva and the eyelid, which is often what we are interested in extracting for describing the shape of eyes. Thus, semi-supervision seems to be a natural way to allow the human expert to label the to-be-detected contours on a few examples, so as to convey the *true* contour that is of real interests for the application at hand.

## 3 Semi-supervised Least-Squares Congealing

First we will describe the basic concept and objective function of the conventional semi-supervised least-square congealing (SLSC) by using image warping [1].

Congealing approaches operate on an ensemble of  $K$  unlabeled images  $\mathbf{I} = \{\mathbf{I}_i\}_{i \in [1, K]}$ , each with an unknown parameter  $\mathbf{p}_i$ , such as the landmark set in [1], that is to be estimated. Semi-supervised congealing also assumes there is a small set of  $\tilde{K}$  labeled images  $\tilde{\mathbf{I}} = \{\tilde{\mathbf{I}}_n\}_{n \in [1, \tilde{K}]}$ , each with a known parameter  $\tilde{\mathbf{p}}_n$ . We denote the collection of all unknown parameters with  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_K]$ . The goal of SLSC is to estimate  $\mathbf{P}$  by minimizing a cost function defined on the entire ensemble:

$$\varepsilon(\mathbf{P}) = \sum_{i=1}^K \varepsilon_i(\mathbf{p}_i). \quad (1)$$

The total cost is the summation of the cost of each unlabeled image  $\varepsilon_i(\mathbf{p}_i)$ :

$$\varepsilon_i(\mathbf{p}_i) = \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K \|f(\mathbf{I}_j, \mathbf{p}_j) - f(\mathbf{I}_i, \mathbf{p}_i)\|^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} \|f(\tilde{\mathbf{I}}_n, \tilde{\mathbf{p}}_n) - f(\mathbf{I}_i, \mathbf{p}_i)\|^2, \quad (2)$$

where  $f(\mathbf{I}, \mathbf{p})$  is the feature representation of image  $\mathbf{I}$  evaluated at  $\mathbf{p}$ . Hence,  $\varepsilon_i(\mathbf{p}_i)$  equals the summation of the pairwise feature difference between  $\mathbf{I}_i$  and all the other images in the ensemble, including both the unlabeled images (the 1<sup>st</sup> term of Eqn. 2) and the labeled image (the 2<sup>nd</sup> term of Eqn. 2).

In [1], the feature representation is defined as,

$$f(\mathbf{I}, \mathbf{p}) \doteq \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p})), \quad (3)$$

where  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  is a warping function that takes  $\mathbf{x}$ , which is a collection of pixel coordinates within the common rectangle region, as input, and outputs the corresponding pixel coordinates in the coordinate space of image  $\mathbf{I}$ . Given this warping function,  $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$  denotes the corresponding warped image vector obtained by bilinear interpolation of the image  $\mathbf{I}$  using the warped coordinates  $\mathbf{W}(\mathbf{x}; \mathbf{p})$ . Note that in [1],  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  is a simple 6-parameter affine warp, rather than a complex non-rigid warp such as the piecewise affine warp [4]. This is due to the high dimensionality in the non-rigid warp, as well as the needs to optimize  $\mathbf{p}$  for all images simultaneously. Hence, by applying affine-warp-based optimization multiple times, each at a different rectangle region with decreasing size, the non-rigid natural of the warp can be approximated.

Since the total cost  $\varepsilon(\mathbf{P})$  is difficult to optimize directly, [1] chooses to iteratively minimize the individual cost  $\varepsilon_i(\mathbf{p}_i)$  for each  $\mathbf{I}_i$ . The well-known inverse warping technique [20] is utilized and after taking the first order Taylor expansion, Eqn. 2 can be simplified to:

$$\frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K \|\mathbf{b}_j + \mathbf{c}_j \Delta \mathbf{p}_i\|^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} \|\tilde{\mathbf{b}}_n + \tilde{\mathbf{c}}_n \Delta \mathbf{p}_i\|^2, \quad (4)$$

where

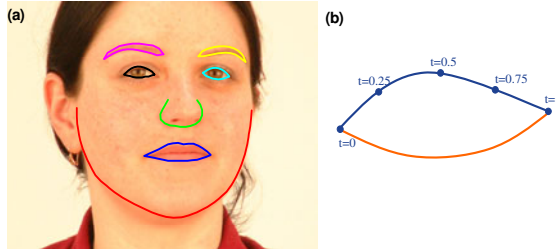
$$\mathbf{b}_j = f(\mathbf{I}_j, \mathbf{p}_j) - f(\mathbf{I}_i, \mathbf{p}_i), \quad \mathbf{c}_j = \frac{\partial f(\mathbf{I}_j, \mathbf{p}_j)}{\partial \mathbf{p}_j}. \quad (5)$$

The least-square solution of Eqn. 4 can be obtained by setting the partial derivative of Eqn. 4 with respect to  $\Delta \mathbf{p}_i$  to be equal to zero. We have:

$$\Delta \mathbf{p}_i = - \left[ \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K \mathbf{c}_j^T \mathbf{c}_j + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} \tilde{\mathbf{c}}_n^T \tilde{\mathbf{c}}_n \right]^{-1} \left[ \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K \mathbf{c}_j^T \mathbf{b}_j + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} \tilde{\mathbf{c}}_n^T \tilde{\mathbf{b}}_n \right]. \quad (6)$$

## 4 Facial Contour Congealing

In this section, we will present our facial contour congealing approach in detail. Three key technical components will be covered: parametric curve representation, curve descriptor, and contour congealing.



**Fig. 2.** (a) The entire facial shape is described by contours on 7 facial components; (b) The closed contour (such as the eye) is represented by two connected parametric curves, where each curve's parameter can be estimated via curve fitting on labeled landmarks (5 landmarks in this case).

#### 4.1 Parametric Curve Representation

In computer vision, it has been very popular to use a set of landmarks to describe the shape of an object by placing the landmarks along the object contour, such as the Point Distribution Model (PDM) applied to faces. However, there are disadvantages to using the landmark representation. First, an excessive number of landmarks are needed in order to approximate the true contour of facial images, especially for high quality images. Second, for the semi-supervised congealing application, little constraint can be applied on the distribution of landmarks since there are very few labeled images, which poses a challenge for landmark estimation on unlabeled images.

In this paper, we propose to use a parametric curve representation to describe the facial contour. As shown in Fig. 2(b), we use two parametric curves to represent the closed contour of one of the seven facial components, such as eye. For simplicity of notation, we will initially focus on one of the two curves, which covers half of the contour. A 2D parametric curve is defined by the  $n$ -order polynomials:

$$x(t) = p_{x,n}t^n + p_{x,n-1}t^{n-1} \cdots + p_{x,1}t + p_{x,0}, \quad (7)$$

$$y(t) = p_{y,n}t^n + p_{y,n-1}t^{n-1} \cdots + p_{y,1}t + p_{y,0}, \quad (8)$$

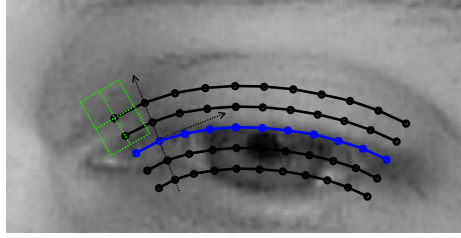
where usually we consider  $t \in [0, 1]$ , and the collection of coefficients,

$$\mathbf{p} = [\mathbf{p}_x \ \mathbf{p}_y]^T = [p_{x,n} \ p_{x,n-1} \ \cdots \ p_{x,1} \ p_{x,0} \ p_{y,n} \ p_{y,n-1} \ \cdots \ p_{y,1} \ p_{y,0}]^T, \quad (9)$$

is called the *curve parameter*, which fully describes the shape of the curve. Given a known  $\mathbf{p}$ , we can generate any number of points on the curve by varying  $t$ .

In practice, when we manually label face images, we label landmarks rather than the curve directly. Suppose there are  $m$  landmarks being manually labeled along the contour, we have:

$$\mathbf{x} = [x(t_1) \ y(t_1) \ \cdots \ x(t_m) \ y(t_m)]^T = \mathbf{T}\mathbf{p}, \quad (10)$$



**Fig. 3.** An array of point coordinates are computed within the band following the target curve. Appearance information, such as pixel intensity or HOG, can be extracted from these coordinates and form a curve descriptor for the target curve.

where

$$\mathbf{T} = \begin{bmatrix} t_1^n & t_1^{n-1} & \dots & t_1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & t_1^n & t_1^{n-1} & \dots & t_1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ t_m^n & t_m^{n-1} & \dots & t_m & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & t_m^n & t_m^{n-1} & \dots & t_m & 1 \end{bmatrix}. \tag{11}$$

By assuming the landmarks are evenly spaced, we have  $[t_1, t_2, \dots, t_m] = [0, \frac{1}{m-1}, \dots, 1]$ . Hence, the curve parameter can be directly computed from the landmark set:

$$\mathbf{p} = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T \mathbf{x}. \tag{12}$$

### 4.2 Curve Descriptor

Having introduced the mapping between the curve parameter and the landmark set, we will present our method to extract the appearance feature for a curve, which is called *curve descriptor*, given the known curve parameter.

For landmark-based shape representation, e.g. Active Shape Model (ASM) [21], the appearance information is often extracted within a small rectangle region around the landmark. Similarly, for curve-based representation, the curve descriptor will be extracted from a *band*-like region along the curve.

As shown in Fig. 3, let us denote the  $U$  points along the central target curve as  $\{[\check{x}_{u,0}, \check{y}_{u,0}]\}_{u=1, \dots, U}$ , where  $[\check{x}_{u,0}, \check{y}_{u,0}] = [x(t_u), y(t_u)]$ . We can allocate  $V$  synthetic curves on both sides of the target curve, where the distance between any neighboring curves is  $r$ . Specifically, for the  $u^{th}$  point on the curve, we have a point  $[\check{x}_{u,v}, \check{y}_{u,v}]$  on its normal direction with a distance  $|v|r$ , which is then located on the  $v^{th}$  synthetic curve,

$$\begin{bmatrix} \check{x}_{u,v} \\ \check{y}_{u,v} \end{bmatrix} = \begin{bmatrix} x(t_u) - vrsin\theta_u \\ y(t_u) + vrcos\theta_u \end{bmatrix}, \tag{13}$$

where  $\theta_u$  is the tangent angle for the  $u^{th}$  point on the curve:

$$\theta_u = \arctan\left(\frac{dy}{dx}\Big|_{t_u}\right) = \arctan\left(\frac{\mathbf{T}'_u \mathbf{p}_y}{\mathbf{T}'_u \mathbf{p}_x}\right), \tag{14}$$

and  $\mathbf{T}'_u$  is the derivative of polynomial evaluated at  $t_u$ :

$$\mathbf{T}'_u = [nt_u^{n-1} (n-1)t_u^{n-2} \dots 1 \ 0]. \tag{15}$$

Hence, with a set of point coordinates  $\check{\mathbf{x}} = \{[\check{x}_{u,v}, \check{y}_{u,v}]\}_{u=1, \dots, U, v=-V, \dots, V}$ , as well as their corresponding angles  $\theta = \{\theta_u\}_{u=1, \dots, U}$ , we can extract the curve descriptor. The simplest descriptor is to use the pixel intensity evaluated at  $\check{\mathbf{x}}$ , i.e.,  $f(\mathbf{I}, \mathbf{p}) \doteq \mathbf{I}(\check{\mathbf{x}})$ . Motivated by the work of [6, 22, 23], we can also use the powerful HOG feature as the curve descriptor:

$$f(\mathbf{I}, \mathbf{p}) \doteq \mathbf{h}(\check{\mathbf{x}}, \theta) = [\hat{\mathbf{h}}(\check{x}_{u,v}, \check{y}_{u,v}, \theta_u)]_{u=1, \dots, U, v=-V, \dots, V}, \tag{16}$$

which is a concatenation of  $U(2V+1)$   $L^2$ -normalized HOG vectors, each centered at  $[\check{x}_{u,v}, \check{y}_{u,v}]$  with angle  $\theta_u$ . Note that the HOG feature we employ makes use of the tangent angle  $\theta$ . Hence it will better capture the appearance information along the curve, as well as on either side of the curve.

### 4.3 Contour Congealing

With the presentation on contour representation and curve descriptor, we now introduce how to conduct contour congealing for an ensemble of facial images. The basic problem setup is the same as the SLSC in Section 3. That is, given the unlabeled image set  $\{\mathbf{I}_i\}$  and its initial label  $\{\mathbf{p}'_i\}$ , as well as a small number of labeled images  $\{\tilde{\mathbf{I}}_n\}$  and their known labels  $\{\tilde{\mathbf{p}}_n\}$ , we need to estimate the true curve parameters  $\{\mathbf{p}_i\}$ .

In this work, our semi-supervised contour congealing is applied on each of the seven components independently. Notice that 5 out of the 7 components are closed contours, where two curve functions are needed to represent the entire contour. In contrast to the SLSC in Section 3, now we face a new challenging problem of congealing two sets of curve parameters simultaneously, where simply applying Eqn. 2 is not sufficient.

By denoting  $\mathbf{p}^1$  and  $\mathbf{p}^2$  as the curve parameters for the top curve and bottom curve respectively, we can utilize one simple geometric constraint. That is, the points on both ends of the 1<sup>st</sup> curve should overlap with those of the 2<sup>nd</sup> curve. With that, our semi-supervised congealing for a closed contour utilizes the following objective function:

$$\varepsilon_i(\mathbf{p}_i^1, \mathbf{p}_i^2) = \varepsilon_i(\mathbf{p}_i^1) + \varepsilon_i(\mathbf{p}_i^2) + \beta \|\mathbf{x}_i^1 - \mathbf{x}_i^2\|^2, \tag{17}$$

where  $\mathbf{x}_i^1 = \mathbf{T}^{01} \mathbf{p}_i^1$ ,  $\mathbf{x}_i^2 = \mathbf{T}^{01} \mathbf{p}_i^2$ , and  $\mathbf{T}^{01}$  is a sub-matrix of  $\mathbf{T}$  including its first two rows and last two rows. This objective function is basically the summation of the error terms from two curves, and their geometric constraint weighted by  $\beta$ .

By employing inverse warping technique [20] and similar simplification as Eqn. 4, we have:

$$\begin{aligned} \varepsilon_i(\Delta \mathbf{p}_i^1, \Delta \mathbf{p}_i^2) &= \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K (\|\mathbf{b}_j^1 + \mathbf{c}_j^1 \Delta \mathbf{p}_i^1\|^2 + \|\mathbf{b}_j^2 + \mathbf{c}_j^2 \Delta \mathbf{p}_i^2\|^2) + \\ \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\|\tilde{\mathbf{b}}_n^1 + \tilde{\mathbf{c}}_n^1 \Delta \mathbf{p}_i^1\|^2 + \|\tilde{\mathbf{b}}_n^2 + \tilde{\mathbf{c}}_n^2 \Delta \mathbf{p}_i^2\|^2) &+ \beta \|\mathbf{x}_i^1 - \mathbf{x}_i^2 - \mathbf{e}_i(\Delta \mathbf{p}_i^1 - \Delta \mathbf{p}_i^2)\|^2, \end{aligned} \quad (18)$$

where  $\mathbf{e}_i = \frac{\partial \mathbf{x}_i^1}{\partial \mathbf{p}_i^1} = \frac{\partial \mathbf{x}_i^2}{\partial \mathbf{p}_i^2} = \mathbf{T}^{01}$ , and  $\mathbf{b}_j^*$  and  $\mathbf{c}_j^*$  can be defined similarly as Eqn. 5.

The curve parameter updates  $\Delta \mathbf{p}_i^1$  and  $\Delta \mathbf{p}_i^2$  can be estimated by solving a linear equation system as:

$$\begin{cases} \frac{\partial \varepsilon_i(\Delta \mathbf{p}_i^1, \Delta \mathbf{p}_i^2)}{\partial \Delta \mathbf{p}_i^1} = 0, \\ \frac{\partial \varepsilon_i(\Delta \mathbf{p}_i^1, \Delta \mathbf{p}_i^2)}{\partial \Delta \mathbf{p}_i^2} = 0. \end{cases} \quad (19)$$

Substituting Eqn. 18 to Eqn. 19, we have:

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B} \\ \mathbf{B} & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{p}_i^1 \\ \Delta \mathbf{p}_i^2 \end{bmatrix} = - \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}, \quad (20)$$

where

$$\mathbf{A}_1 = \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K (\mathbf{c}_j^1)^T \mathbf{c}_j^1 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^1)^T \tilde{\mathbf{c}}_n^1 + \beta \mathbf{e}_i^T \mathbf{e}_i, \quad (21)$$

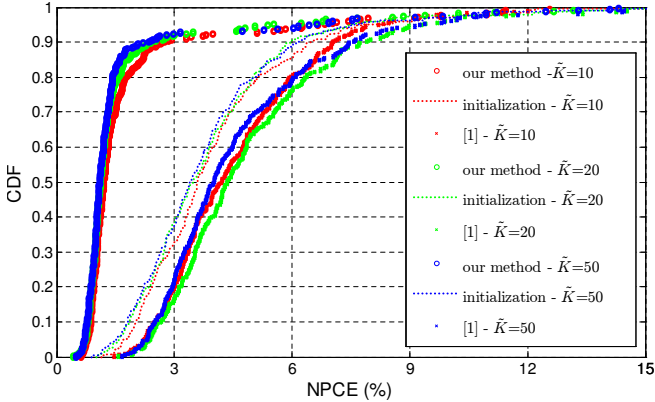
$$\mathbf{A}_2 = \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K (\mathbf{c}_j^2)^T \mathbf{c}_j^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^2)^T \tilde{\mathbf{c}}_n^2 + \beta \mathbf{e}_i^T \mathbf{e}_i, \quad (22)$$

$$\mathbf{B} = -\beta \mathbf{e}_i^T \mathbf{e}_i, \quad (23)$$

$$\mathbf{C}_1 = \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K (\mathbf{c}_j^1)^T \mathbf{b}_j^1 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^1)^T \tilde{\mathbf{b}}_n^1 - \beta \mathbf{e}_i^T (\mathbf{d}_i^1 - \mathbf{d}_i^2), \quad (24)$$

$$\mathbf{C}_2 = \frac{1-\alpha}{K-1} \sum_{j=1, j \neq i}^K (\mathbf{c}_j^2)^T \mathbf{b}_j^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^2)^T \tilde{\mathbf{b}}_n^2 + \beta \mathbf{e}_i^T (\mathbf{d}_i^1 - \mathbf{d}_i^2). \quad (25)$$

The above solution is straightforward to implement as long as we know how to compute  $\mathbf{b}_j^*$  and  $\mathbf{c}_j^*$ , among which  $\frac{\partial f(\mathbf{L}, \mathbf{p})}{\partial \mathbf{p}}$  will likely take the most effort to compute. Hence, from now on we will focus on the computation of  $\frac{\partial f(\mathbf{L}, \mathbf{p})}{\partial \mathbf{p}}$  when the curve descriptor is the HOG feature. For the case of the intensity feature,  $\frac{\partial f(\mathbf{L}, \mathbf{p})}{\partial \mathbf{p}}$  is relatively easier and will be omitted from this discussion.



**Fig. 4.** Performances of the contours on two eyes using our algorithm, the baseline and initialization, when the number of labeled images  $\bar{K}$  is 10, 20, and 50

Note that our HOG feature is a  $L^2$ -normalized version,  $\hat{\mathbf{h}} = \frac{\mathbf{h}}{\|\mathbf{h}\|_2}$ , due to the proven superior performance over the non-normalized version [6]. Hence,

$$\frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{p}} = \frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{p}} \tag{26}$$

$$= \left( \frac{\mathbf{I}^{32}}{\|\mathbf{h}\|_2} - \frac{1}{(\|\mathbf{h}\|_2)^{3/2}} \mathbf{h} \mathbf{h}^T \right) \left( \frac{\partial \mathbf{h}}{\partial \check{x}_{u,v}} \frac{\partial \check{x}_{u,v}}{\partial \mathbf{p}} + \frac{\partial \mathbf{h}}{\partial \check{y}_{u,v}} \frac{\partial \check{y}_{u,v}}{\partial \mathbf{p}} + \frac{\partial \mathbf{h}}{\partial \theta_u} \frac{\partial \theta_u}{\partial \mathbf{p}} \right), \tag{27}$$

where  $\mathbf{I}^{32}$  is a  $32 \times 32$  identity matrix,

$$\frac{\partial \check{x}_{u,v}}{\partial \mathbf{p}} = \frac{\partial \check{x}_{u,0}}{\partial \mathbf{p}} - v r \cos \theta_u \frac{\partial \theta_u}{\partial \mathbf{p}}, \tag{28}$$

and

$$\frac{\partial \theta_u}{\partial \mathbf{p}} = \frac{1}{1 + (\tan \theta_u)^2} \frac{\partial \frac{\mathbf{T}'_u \mathbf{p}_y}{\mathbf{T}'_u \mathbf{p}_x}}{\partial \mathbf{p}} \tag{29}$$

$$= \frac{1}{1 + (\tan \theta_u)^2} \left[ -\frac{\mathbf{T}'_u \mathbf{p}_y}{(\mathbf{T}'_u \mathbf{p}_x)^2} \mathbf{T}'_u \ 0 \ \frac{1}{\mathbf{T}'_u \mathbf{p}_x} \mathbf{T}'_u \ 0 \right]. \tag{30}$$

The partial derivatives  $\frac{\partial \mathbf{h}}{\partial \check{x}_{u,v}}$ ,  $\frac{\partial \mathbf{h}}{\partial \check{y}_{u,v}}$ , and  $\frac{\partial \mathbf{h}}{\partial \theta_u}$  can be computed using the definition of derivative in the discrete case, i.e.,  $\frac{\partial \mathbf{h}}{\partial \check{x}_{u,v}} = \mathbf{h}(\check{x}_{u,v}, \check{y}_{u,v}, \theta_u) - \mathbf{h}(\check{x}_{u,v} - 1, \check{y}_{u,v}, \theta_u)$ . Similar ways of computing  $\frac{\partial \mathbf{h}}{\partial x}$  and  $\frac{\partial \mathbf{h}}{\partial y}$  have been used in [22].

For the case of open facial contour, such as nose and chin, we use the first term of Eqn. 17 as the objective function. Its solution is a simplified case of the above derivation, and hence will be omitted here.

## 5 Experimental Results

In this section, we will present the extensive experiments that demonstrate the capability of our proposed algorithm. For our experiments, we choose a subset of 350 images from the publicly-available PUT face database [24], which exhibits moderate variation in pose and facial expression (Fig. 1). The entire image set is partitioned into two sets: one with 300 images is used as the unlabeled image ensemble  $\mathbf{I}$ , the other with 50 images will be randomly chosen as the labeled image set  $\tilde{\mathbf{I}}$ . All images have manually labeled ground-truth on the facial contours of 7 components (Fig. 2). For example, the contour of an eye is labeled with 20 landmarks. There are 166 total landmarks labeled for all 7 contours. This ground-truth will not only provide the known curve parameter  $\tilde{\mathbf{p}}_n$  for labeled image  $\tilde{\mathbf{I}}_n$  (via Eqn. 12), but also be used in quantitative evaluation of the performance.

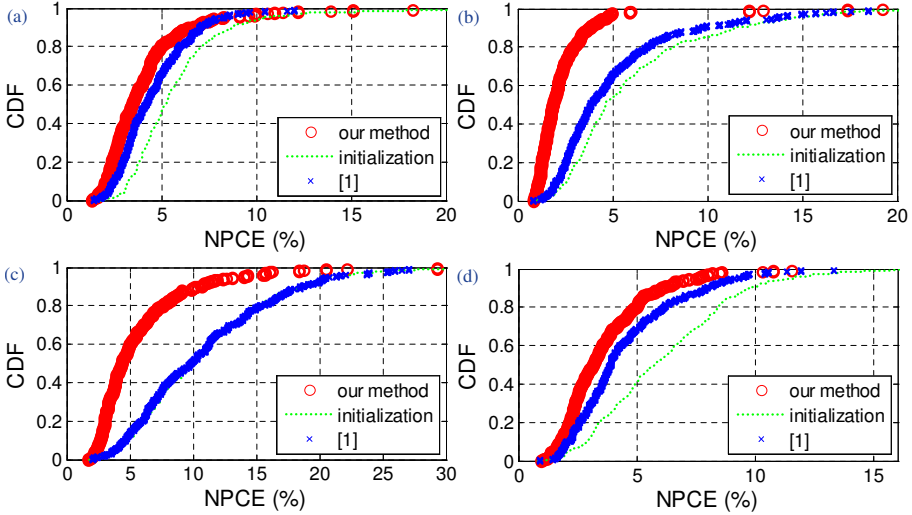
Since the very recent work of Tong *et al.* [1] is the most relevant to ours, it is chosen as the baseline approach for comparison. We have implemented both algorithms in Matlab and ensure they are tested under the same condition. Although PUT is a high quality face database, we downsample the face size to be around 70 pixels eye-to-eye, mostly based on the concern that the efficiency of the baseline algorithm largely depends on the average face size. For the labeled set, both algorithms have their  $\tilde{\mathbf{p}}$  parameters computed from the manually labeled 166 landmarks per image. For the unlabeled set, both algorithms use the PittPatt face detector [25] to compute the initial  $\mathbf{p}$ , by placing an average set of landmarks/contours (see the top-right of Fig. 1), which is computed from the small set of labeled images, within the detected face rectangle. This is a valid initialization since face detection is almost a commodity.

For our algorithm, once the estimation of curve parameters is completed, we compute the average of the distances between each ground-truth landmark and the estimated curve, and then divide it by the distance between the two eye centers. This quantitative evaluation is called *Normalized Point to Curve Error* (NPCE), and is expressed as a percentage. We also compute NPCE for the baseline because a curve function can be fitted to the estimated landmarks via the baseline algorithm.

### 5.1 Performance Comparison

We will first present the performance comparison between our algorithm and the baseline approach. For each unlabeled image in our algorithm, once the average contour is placed within the face detection rectangle, we have the initial curve parameters for all seven components. Then the contour congealing is conducted on each component independently. Note that we only use the very basic face detection functionality and no additional landmark detection, such as eye and nose, is conducted using the PittPatt SDK. Hence, it is obvious that face detection can have quite a large variation on localization, especially for smaller components. However, our algorithm does surprisingly well in handling this real-world challenging initialization and congealing all components independently. Of course, one potential future work is to use the better-congealed components, and





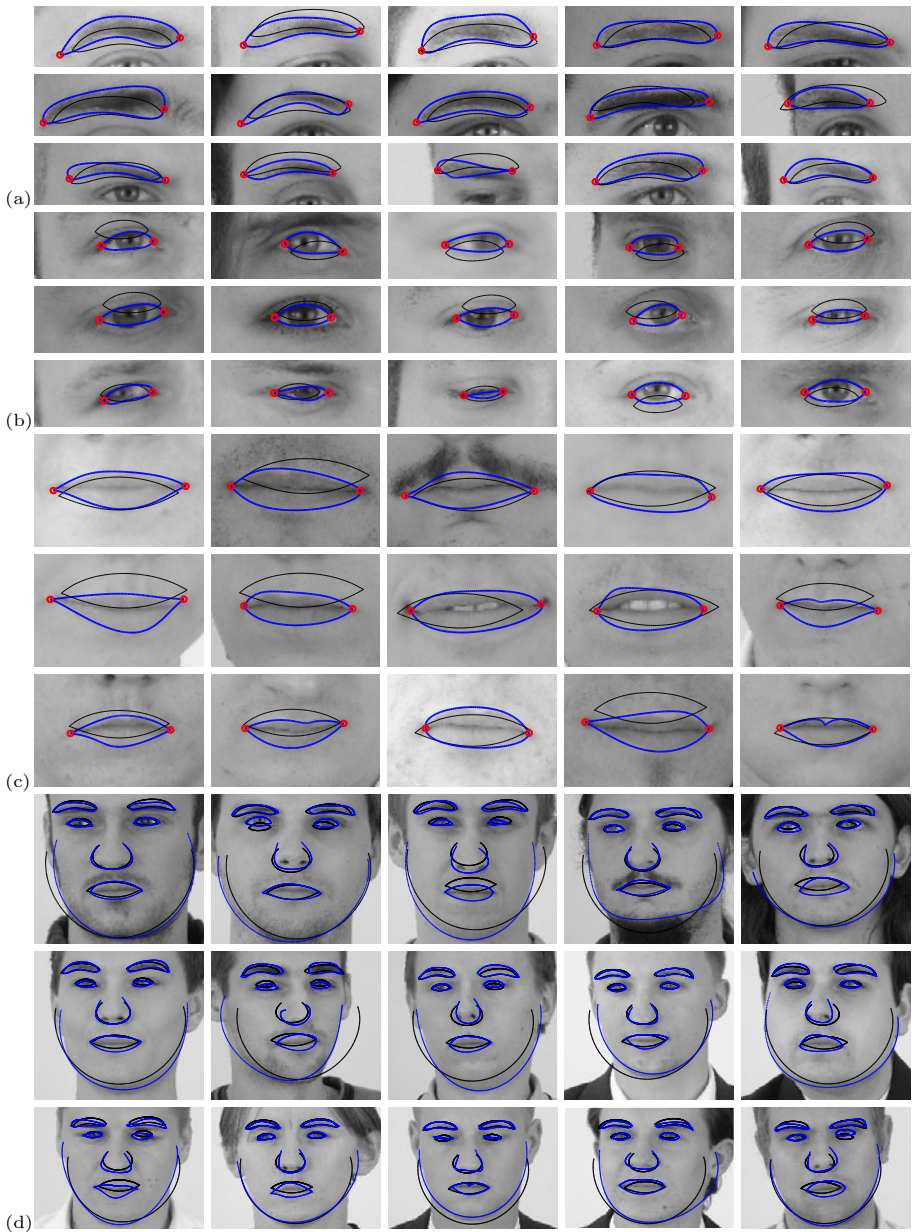
**Fig. 5.** Comparison of our algorithm, the baseline and initialization ( $\tilde{K} = 10$ ) for (a) two eyebrows, (b) mouth, (c) chin, (d) nose

global spatial distribution of components learned from labeled data, to produce a better initialization for other components.

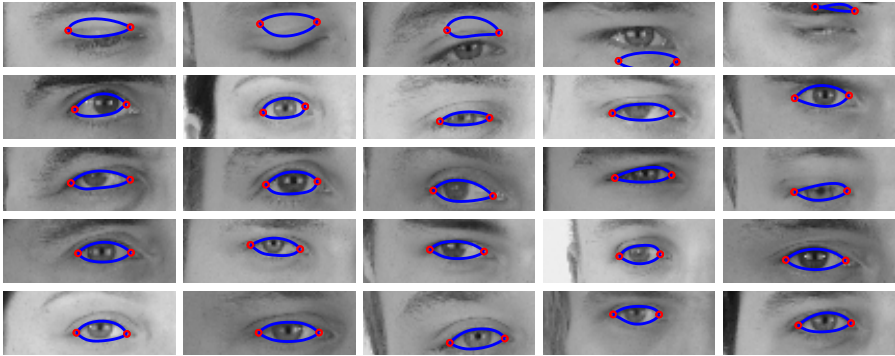
For both algorithms, we perform three sets of experiments, each with a different number of labeled images,  $\tilde{K}=10, 20$ , and  $50$ . For all components in our algorithm, we use  $4^{th}$ -order polynomials ( $n=4$ ) in the curve function, and the  $2 \times 2$  cell 8-bin HOG feature for the curve descriptor, where  $V \in [3, 5]$  and  $r \in [1, 2]$  for various components. We fix  $\alpha = 0.5$  and  $\beta = 50$  throughout the experiments.

To better understand the system performance, we plot the comparison for two eye components in Fig. 4. The cumulative distribution function (CDF) of NPCE is plotted for the results of our algorithm, the baseline, and the initialization via face detection. It is clear that our algorithm improves the initialization with a large margin, while the baseline performs slightly worse than the initialization. We attribute this worse performance of the baseline to the pose variation in the data, which makes the image warping and progressive affine approximation less likely to work well. Note that for our algorithm, more than 90% of the unlabeled images have the final eye localization error less than 2% of eye-to-eye distance. For the right eye, it takes our algorithm 13 – 15 iterations (about 3.5 hours on the conventional PC) to converge for the entire test set when  $\tilde{K}$  is 10 or 20.

In comparing the results with various  $\tilde{K}$ , we can see that our approach at  $\tilde{K} = 10$  is almost as good as when  $\tilde{K} = 50$ . This is a very important property since it means our approach can be used with a very small set of labeled images. The similar property is also observed in the comparison of other components. Hence, due to limited space, we show the results of other components only



**Fig. 6.** The initialization and results of our algorithm on various facial components: (a) left eyebrow ( $\tilde{K}=20$ ), (b) left eye ( $\tilde{K}=50$ ), (c) mouth ( $\tilde{K}=10$ ), and (d) whole face ( $\tilde{K}=10$ ). It can be seen that some of the images, especially those with background displayed, are of faces with noticeable pose variation. Notice the large amount of shape variation exhibited in the data that can be handled by our algorithm.

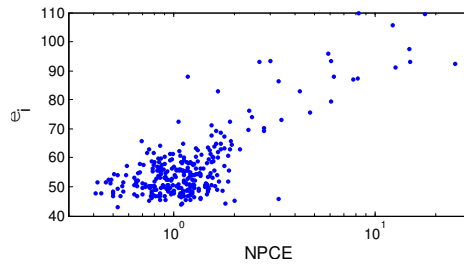


**Fig. 7.** The confidence of labeling the eye component increases from the top row to bottom row. We observe that almost all failed cases can be found in the category with lowest confidence score.

when  $\tilde{K} = 10$  in Fig. 5. Again, for all the remaining components, our algorithm performs substantially better than the baseline, which also improves over the initialization except the chin contour. We attribute the improvement of our algorithm to three reasons: 1) the partition scheme and using a set of affine warps to approximate non-rigid deformation of [1] pose limitation on the accuracy; 2) the feature extracted along the curve better describes the appearance information than the feature in the partitioned rectangle of [1]; 3) the HOG feature is more suitable for localization than the intensity feature in [1]. We also illustrate the congealing results of our approach on various components in Fig. 6.

## 5.2 Labeling Confidence

Knowing when an algorithm does not converge is often as important as overall algorithm performance. This is especially true for semi-supervised algorithms. Hence, a confidence score is desirable for practical applications in order to evaluate the quality of labeling without ground truth. For this we use  $\varepsilon_i(\mathbf{p}_i^1, \mathbf{p}_i^2)$  in Eqn. 17. A smaller  $\varepsilon_i$  indicates a higher-confidence in labeling. By partitioning the 300 confidence scores into 5 bins, Fig. 7 shows labeled left eye component from the lowest 20% to the highest 20% confidence scores, in our 300-image ensemble ( $\tilde{K} = 10$ ). Fig. 8 also illustrates the distribution of the estimated  $\varepsilon_i$  versus the actual labeling error represented by the NPCE for the left eye component. With the increase of the  $\varepsilon_i$ , the landmark labeling error increases significantly. Hence, it is clear that this confidence score is indicative of labeling performance. The linear correlation between  $\varepsilon_i$  and NPCE can also be shown by the computed Pearson correlation coefficient, which is 0.715. Similar phenomena have been observed for experiments on other facial components. In practice, after labeling, one can use this confidence score to select only high-confident samples for a training set, or to select low-confident samples for other appropriate additional processing.



**Fig. 8.** The correlation between the labeling confidence ( $\varepsilon_i$ ) and actual labeling error (NPCE). The Pearson correlation coefficient between these two variables is 0.715.

## 6 Conclusions

Real-world objects can exhibit a large amount of shape deformation on 2D images due to intra-object variability, object motion, and camera viewpoint. Rather than the conventional landmark-based representation, we propose to use curve functions to describe the facial contour. We demonstrate a complete system that is able to simultaneously align facial contour for a large set of unlabeled images with face detection results, given a few labeled images. Extensive experiments demonstrate that our system has achieved much more accurate labeling results compared to the previous state-of-the-art approach on face images with moderate changes in pose and expression.

## References

1. Tong, Y., Liu, X., Wheeler, F.W., Tu, P.: Automatic facial landmark labeling with minimal supervision. In: CVPR (2009)
2. Miller, E., Matsakis, N., Viola, P.: Learning from one example through shared densities on transforms. In: CVPR, vol. 1, pp. 464–471 (2000)
3. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. *IEEE T-PAMI* 23, 681–685 (2001)
4. Matthews, I., Baker, S.: Active appearance models revisited. *IJCV* 60, 135–164 (2004)
5. Liu, X.: Discriminative face alignment. *IEEE T-PAMI* 31, 1941–1954 (2009)
6. Dalal, N., Triggs, W.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
7. Vetter, T., Jones, M.J., Poggio, T.: A bootstrapping algorithm for learning linear models of object classes. In: CVPR, pp. 40–46 (1997)
8. Learned-Miller, E.: Data driven image models through continuous joint alignment. *IEEE T-PAMI* 28, 236–250 (2006)
9. Cox, M., Sridharan, S., Lucey, S., Cohn, J.: Least squares congealing for unsupervised alignment of images. In: CVPR (2008)
10. Balci, S., Golland, P., Shenton, M., Wells, W.: Free-form B-spline deformation model for groupwise registration. In: MICCAI, pp. 23–30 (2007)
11. Baker, S., Matthews, I., Schneider, J.: Automatic construction of active appearance models as an image coding problem. *IEEE T-PAMI* 26, 1380–1384 (2004)

12. Kokkinos, I., Yuille, A.: Unsupervised learning of object deformation models. In: ICCV (2007)
13. Cootes, T., Twining, C., Petrovic, V., Schestowitz, R., Taylor, C.: Groupwise construction of appearance models using piece-wise affine deformations. In: BMVC, vol. 2, pp. 879–888 (2005)
14. Cristinacce, D., Cootes, T.: Facial motion analysis using clustered shortest path tree registration. In: Proc. of the 1st Int. Workshop on Machine Learning for Vision-based Motion Analysis with ECCV (2008)
15. Torre, F., Nguyen, M.: Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment. In: CVPR (2008)
16. Langs, G., Donner, R., Peloschek, P., Horst, B.: Robust autonomous model learning from 2D and 3D data sets. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 968–976. Springer, Heidelberg (2007)
17. Saragih, J., Goecke, R.: A nonlinear discriminative approach to AAM fitting. In: ICCV (2007)
18. Sidorov, K., Richmond, S., Marshall, D.: An efficient stochastic approach to groupwise non-rigid image registration. In: CVPR (2009)
19. Meltzer, J., Soatto, S.: Edge descriptors for robust wide-baseline correspondence. In: CVPR (2008)
20. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. IJCV 56, 221–255 (2004)
21. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models — their training and application. CVIU 61, 38–59 (1995)
22. Liu, X., Yu, T., Sebastian, T., Tu, P.: Boosted deformable model for human body alignment. In: CVPR (2008)
23. Liu, X., Tong, Y., Wheeler, F.W.: Simultaneous alignment and clustering for an image ensemble. In: ICCV (2009)
24. Kasinski, A., Florek, A., Schmidt, A.: The PUT face database. Technical report, Poznan University of Technology, Poznan, Poland (2009)
25. Schneiderman, H.: Learning a restricted Bayesian network for object detection. In: CVPR (2004)

# Cascaded Confidence Filtering for Improved Tracking-by-Detection

Severin Stalder<sup>1</sup>, Helmut Grabner<sup>1</sup>, and Luc Van Gool<sup>1,2</sup>

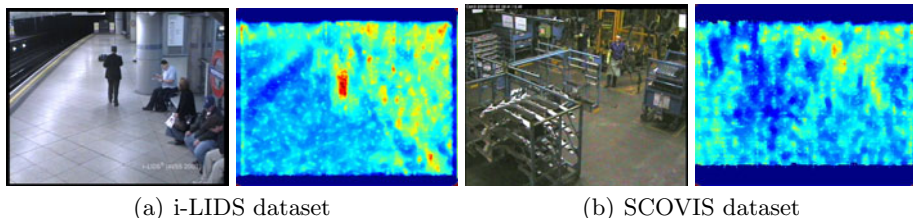
<sup>1</sup> Computer Vision Laboratory, ETH Zurich, Switzerland  
{sstalder, grabner, vangool}@vision.ee.ethz.ch

<sup>2</sup> ESAT - PSI / IBBT, K.U. Leuven, Belgium  
luc.vangool@esat.kuleuven.be

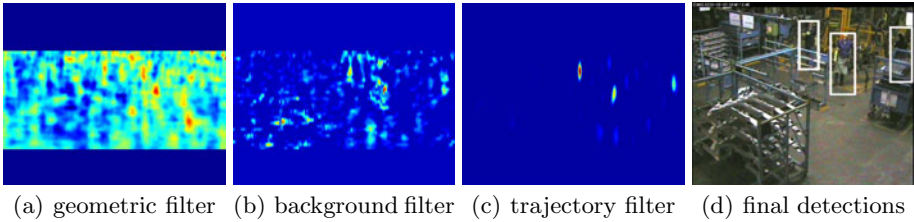
**Abstract.** We propose a novel approach to increase the robustness of object detection algorithms in surveillance scenarios. The cascaded confidence filter successively incorporates constraints on the size of the objects, on the preponderance of the background and on the smoothness of trajectories. In fact, the continuous detection confidence scores are analyzed locally to adapt the generic detector to the specific scene. The approach does not learn specific object models, reason about complete trajectories or scene structure, nor use multiple cameras. Therefore, it can serve as preprocessing step to robustify many tracking-by-detection algorithms. Our real-world experiments show significant improvements, especially in the case of partial occlusions, changing backgrounds, and similar distractors.

## 1 Introduction

Monocular multi-object detection and tracking with static cameras is a challenging, but practically important problem. The task is inherently difficult due to the variability of object and background appearances. Most current methods depend on careful camera placement, as to avoid occlusions, see Fig. 1(a). However, in many settings this is virtually impossible, with deteriorating detection results as a consequence as illustrated in Fig. 1(b).



**Fig. 1.** (a) Typical state-of-the-art methods for object detection perform quite well when applied to current datasets, i.e. the maximum of the detection confidence map clearly corresponds to the fully visible person. (b) However, the detection confidence map is highly ambiguous in more cluttered scenes such as the SCOVIS dataset due to partial occlusions and similar structures in the background.



**Fig. 2.** The proposed *Cascaded Confidence Filter* (CCF) refines the noisy confidence map to robustify tracking-by-detection methods. It combines prior information about the object size (a), predominant background (b) and smooth trajectories (c) in order to detect objects which move smoothly on the ground plane (d). The confidences correspond to the foot-points on the ground plane.

Currently, most object detectors consist of a learned appearance model, e.g., [1,2]. Despite significant, recent improvements, e.g., [3], their accuracy is still far from perfect. Further improvements increasingly come from the analysis of the detection context rather than merely analyzing the object patch, e.g., considering perspective [4]. In fact, context is also beneficial in pre-processing [5] or during learning of object detectors to better match training data to test data. In particular, classifier grids [6] learn separate object detectors at each image location in an on-line manner. However, unsupervised learning of such models might lead to label noise and deterioration of results [7].

Tracking-by-detection methods, e.g., [8,9], apply object detection independently in each frame and associate detections across frames to bridge gaps and to remove false positives. The use of continuous detection confidence scores in combination with thresholded detections and specific object models has been shown to facilitate target association [10]. Furthermore, scene specific knowledge, like entry/ exit zones to initialize trackers, helps to improve tracking results [9]. Yet, in tracking-by-detection with a static camera, one might also benefit from long-term observations of the scene, similar to detection and tracking in early surveillance scenarios which involved background models and change detection, e.g., [11].

We introduce a cascaded filtering of the detector confidence map, coined *Cascaded Confidence Filtering* or CCF. CCF incorporates constraints on the size of the object, on the preponderance of the background and on the smoothness of trajectories. In fact, the results of any sliding window detection algorithm can be improved with CCF in case of a static camera. As an example, Fig. 2 shows the successively refined detector confidence map from Fig. 1(b), finally allowing for detection of all three persons in that scene.

**Contribution.** CCF adapts a generic person detector to a specific scene. In particular, CCF combines a number of the reviewed approaches, trying to inherit their strengths while avoiding their weaknesses.

- CCF refrains from globally thresholding the detections. Instead, the continuous confidence scores of the detector are modeled at each position separately, similar to classifier grid approaches. In contrast to the latter, we rather use a fixed object model and analyze the detection confidences similar to color background modeling. The embedded use of a discriminative object model in background modeling allows to circumvent difficulties with noisy blobs, partial occlusions and different objects with similar shape.
- Detection responses are put into their spatial and temporal context as with tracking-by-detection. But these confidence levels are first filtered on a small spatial and a short temporal scale before reasoning about longer trajectories or scene structure. In fact, the smoothness of trajectories is ensured through a process analogous to vessel filtering in medical imaging. Those additional constraints permit to keep the advantages of an object detector while suppressing detections on background structures.

The refined detection confidence map can then be given to various existing tracking-by-detection frameworks modeling the scene structure and learning specific object models. It is shown that the filtered detections can be associated to long trajectories employing a typical tracking-by-detection method. Conversely, the same tracking-by-detection method performs dismally using the unfiltered detections.

The remainder of the paper is organized as follows. In Sec. 2 we present our CCF approach. Detailed experiments as well as improved object detection and tracking results are shown in Sec. 3. Finally, Sec. 4 suggests further work and concludes the paper.

## 2 Cascaded Confidence Filtering Approach

The input for our algorithm are confidence scores  $S$ , which are proportional to the likelihood that the object of interest appears at a certain position, i.e.,  $P(obj) \propto S$ . More formally, let  $S(I, x, y, s) \in \mathbb{R}$  be the continuous confidence score of the object detector at the center position  $(x, y)$  and scale  $s$  in an image  $I$ . Then, the goal of our method is to successively filter the detection confidence responses by including spatial and temporal context as depicted in Fig. 2 and more detailed in Fig. 3. Summarizing, any object of the object class of interest moving smoothly on a ground plane fulfills the assumptions of the filter steps and their detection scores are enhanced accordingly. Each individual filtering step is described in one of the following subsections.

### 2.1 Geometric Filter

The geometric filter incorporates the assumption that the objects move on a common ground plane restricting their possible size in the image, see second row of Fig. 3. This constraint has already been used either as post-processing in order to suppress inconsistent detections (e.g., 4) or more recently as pre-processing to fit the image better to the training data of the detector, e.g., 5.



Following the latter approach, we only evaluate the object detection scores  $S'$  within appropriate candidate windows

$$P(obj|geometry) \propto S' = S(I, x, y, s) \quad (1)$$

where  $(x, y, s)$  satisfy the geometric constraints.

## 2.2 Background Filter

The second filter benefits from long-term observations as typically provided in surveillance situations. We follow a similar approach as in traditional background modeling. However, instead of modeling the pixel color values as in [11], we model the geometrically filtered confidence scores  $S'$ . Those are modeled using mixture of Gaussians ensuring robustness against environmental changes, e.g., of illumination. In fact, the detector confidence scores  $S'$  are modeled separately at each location as

$$P(S') = \sum_{k=1}^K w_k \eta(S', \mu_k, \sigma_k^2) \quad (2)$$

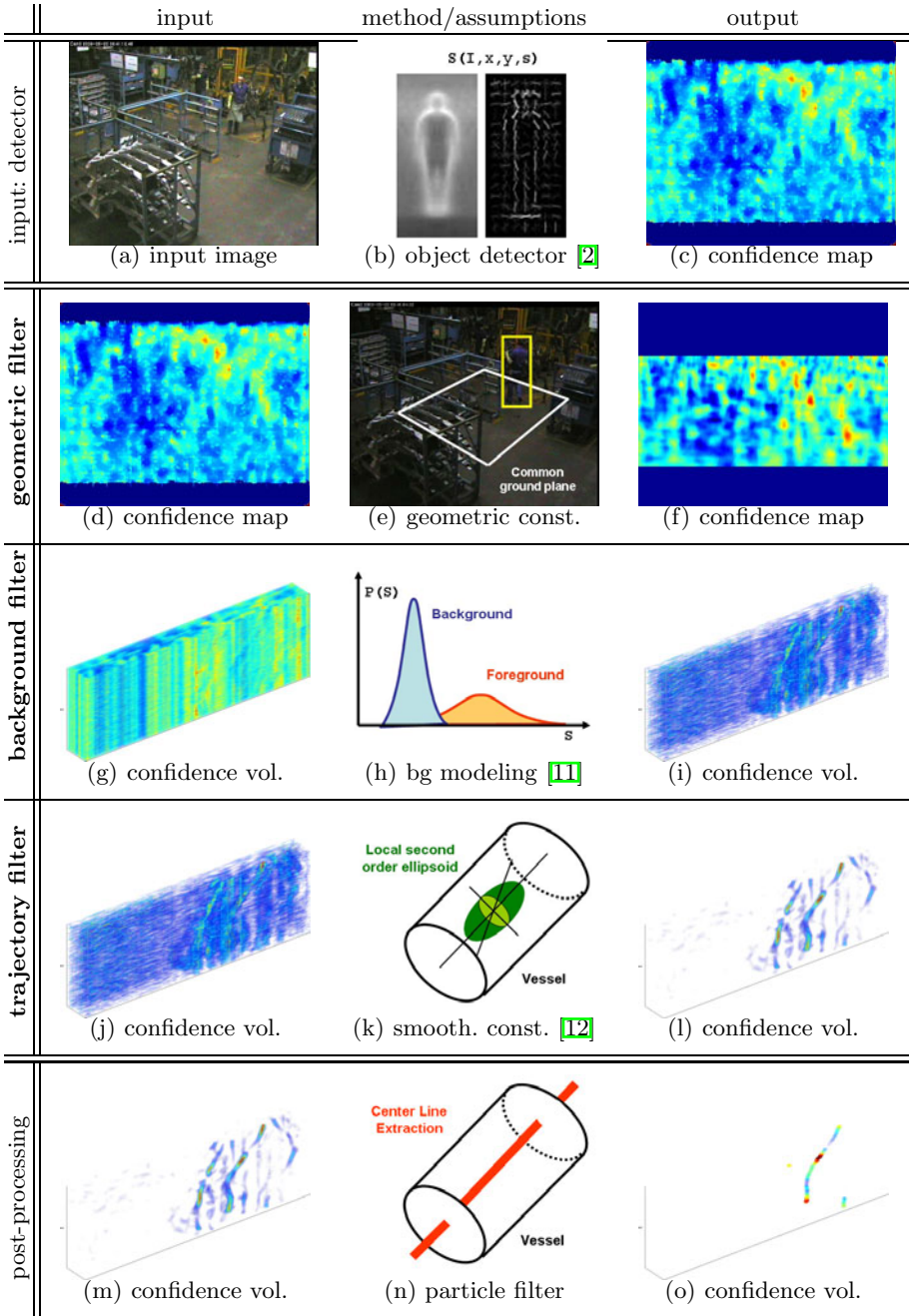
$$\text{where } \eta(S', \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(S'-\mu)^2}{2\sigma^2}}. \quad (3)$$

Updating the  $K$  mixture components and their weights  $w_k$  is done in the same on-line manner as described in [11]. We also use the heuristic assumptions of the latter approach to identify the mixture components belonging to background activities, i.e. by selecting the Gaussian distributions which have the most supporting evidence (mixture weights  $w_k$ ) and the least variance  $\sigma_k^2$ . In fact, the mixture components are sorted according to  $w/\sigma^2$  and the first  $B$  components are selected to model a defined portion  $T$  of the data. Finally, the probability of the object belonging to the foreground class is given by

$$P(obj|geometry, background) \propto S'' = \begin{cases} 1 - \sum_{b=1}^B w_b \eta(S', \mu_b, \sigma_b^2) & \text{if } \forall b : S' > \mu_b \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } B = \arg \min_b \left( \sum_{i=1}^b w_i > T \right). \quad (4)$$

The intuition behind this heuristic is that the variance in the detection confidence of a background activity is smaller than the variance of moving objects of interest. Moreover, the background is generally expected to be present more often than moving objects. In contrast to [11], we only consider confidences higher than the mean of the assumed background distribution, since object presence should always lead to a higher detection confidence even when the background confidence is already quite high.



**Fig. 3.** The detection confidences are successively refined by our cascaded filtering approach incorporating a-priori constraints on scene, objects and trajectories

The proposed background filter can also be seen as *location specific threshold adaptation* for detection. Consequently, (i) systematically occurring false positives are removed, which increases precision, whereas (ii) the sensitivity in other regions (e.g., containing occluders) is increased, which possibly increases recall. An example is depicted in the third row of Fig. 3.

### 2.3 Trajectory Filter

The background filtered confidence map is supposed to suppress static background structures. The trajectory filter is designed to exclude other moving objects which might pass the previous filter step. In fact, there are two observations which are not likely to be correct for other moving objects. Firstly, an object appearance usually causes multiple (similar) responses within a local region in the image (see Fig. 1(a) or the related discussion in 1). Secondly, the confidences should also be continuously changing over time, i.e. the object is not disappearing completely from one frame to the next.

We propose to analyze the volume spanned by the temporal aggregation of the confidence maps. In that volume, we enhance geometrical structures which resemble tubes while suppressing spurious or multiple responses. Therefore we apply the vessel filter approach proposed by Frangi et al. 12 in our context of trajectory filtering. In fact, the approach was originally designed to enhance vascular or bronchial vessel structures in medical data like magnetic resonance images. So a fitted ellipsoid in the defined spatial/ temporal volume  $\Theta$  is used to extract the direction of elongation of high confidences. The approach does analyze the eigenvectors of the Hessian matrix to calculate the principal directions in which the local second order structure of the image can be decomposed. Therefore the eigenvalues  $\lambda_1, \lambda_2, \lambda_3$  (sorted ascend with respect to their absolute value) directly describe the curvature along the vessel. An ideal vessel has corresponding eigenvalues which fulfill the constraints

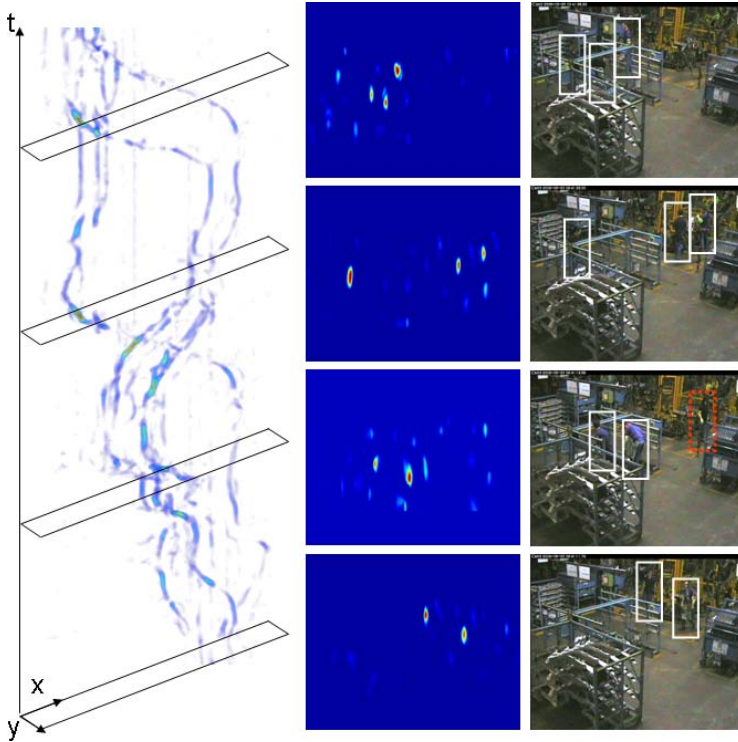
$$|\lambda_1| \approx 0 \text{ and } |\lambda_1| \ll |\lambda_2| \text{ and } \lambda_2 \approx \lambda_3, \quad (5)$$

,i.e., more or less no curvature along the first principle axis and similar (higher) curvature along the other two axes. The sign of the eigenvalues is negative in our case as locations on the trajectories are indicated through higher  $P(obj|geometry, background)$ . The “vesselness” measure is obtained on the basis of all eigenvalues of the Hessian matrix as

$$V(\Theta) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \vee \lambda_3 > 0 \\ \left(1 - e^{-\frac{\lambda_1^2}{2|\lambda_2\lambda_3|\alpha^2}}\right) e^{-\frac{\lambda_2^2}{2\lambda_3^2\beta^2}} \left(1 - e^{-\frac{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}{2c^2}}\right) & \text{otherwise} \end{cases} \quad (6)$$

where  $\alpha, \beta$  and  $c$  are thresholds which control the sensitivity (please refer to 12 for more details). This function provides the probabilistic output

$$P(obj|geometry, background, trajectory) \propto V(\Theta). \quad (7)$$



**Fig. 4.** Left: filtered detector confidence volume by our CCF approach with four transparent planes at selected time instances. Middle: confidence maps for the 4 instances projected onto the ground plane. Right: the corresponding frames overlaid with the detection results. As it can be seen, all the detections at this stage (white) correspond to true positives. There are no false positives among them in this example. Only one person was left undetected (dotted red).

An example is shown in the fourth row of Fig. 3. Additionally, a longer sequence (about half a minute) with corresponding images overlaid with the detections is depicted in Fig. 4.

## 2.4 Post-Processing

After applying the trajectory filter, possible trajectory-like structures are enhanced, but not yet segmented. Inspired by particle filtering methods of [13] for segmentation of coronaries and [10] for multi-object tracking, we use a simple particle filter to resolve ambiguities and output detections, see last row of Fig. 3. This last filtering step has a similar purpose as the common non-maxima suppression, used for instance in [1] to get non-overlapping detections. However, in the case of video streams, one should benefit from the temporal information.

We use a bootstrap filter, where the state of a particle  $\mathbf{x} = (x, y, u, v)$  consists of the 2D position  $(x, y)$  on the ground plane and the velocity components  $(u, v)$ . For prediction, we assume a constant velocity motion model, i.e.,

$$(x, y)_t = (x, y)_{t-1} + (u, v)_{t-1} \cdot \Delta t + \epsilon_{(x,y)} \quad (8)$$

$$(u, v)_t = (u, v)_{t-1} + \epsilon_{(u,v)} \quad (9)$$

where the process noise  $\epsilon_{(x,y)}$ ,  $\epsilon_{(u,v)}$  for each state variable is independently drawn from zero-mean normal distributions.

As observations we use the values from the filtered spatial/ temporal volume. The importance weight  $w^{(i)}$  for each particle  $i$  at time step  $t$  is then described by:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot P(\text{obj}|(x, y)_t, \text{geometry}, \text{background}, \text{trajectory}) \quad (10)$$

From that weight distribution, re-sampling in each time step is performed using a fixed number of  $N$  particles.

For tracklet initialization, the filtered confidence must be above a certain user defined threshold  $\theta_{init}$ .

Furthermore, the particles are gated in order to remove the multi-modality provided by particle filtering if the filtered confidence exceeds another user-defined threshold  $\theta_{gating} \geq \theta_{init}$ .

For each image, the particle with the highest filtered confidence corresponds to the assumed position of the object in this frame. The object can then be detected in the image by mapping it back to the corresponding image coordinates.

### 3 Experimental Results

In principle, the proposed CCF approach can be used with any sliding windows object detector, e.g., to locate cars or faces, as long as the camera remains static. However, our experiments are focused on the task of human detection.

**Datasets.** We use two video sequences for evaluation: (i) the public i-LIDS AB Easy dataset<sup>1</sup> and (ii) our recorded SCOVIS dataset.

The SCOVIS dataset was captured at a workstation in a car manufacturing site. It is challenging due to the industrial working conditions (e.g., sparks and vibrations), difficult structured background (e.g., upright racks, and heavy occlusions of the workers in most parts of the image), and other moving objects (e.g., welding machines and forklifts). Of the total 8 hours of video we evaluated a sequence of 5,000 frames after 5,000 frames used to initialize the background statistics. We manually annotated every  $10^{th}$  frame for evaluation purposes.

#### 3.1 Implementation Details

In this section, we shortly report the details of our implementation. Most sub-parts are publicly available and should make our experiments easily repeatable.

<sup>1</sup> Available at [http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007\\_d.html](http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html), 2010/03/10

**Person detector (HoG).** We use the OpenCV 2.0 implementation of the histogram of oriented gradients (HoG) person detector [22]. Please note that more sophisticated part-based approaches, e.g., [3], could also be used.

**Geometric filter (GF).** We assume one common ground plane in the scene with a flat horizon. The ground plane calibration, i.e., determining the scale of a typical person at each position on the ground plane is done manually. The detector is evaluated on a virtual grid in the image plane with  $32 \times 297$  resolution in the SCOVIS and  $32 \times 375$  resolution in the i-LIDS dataset.

**Background filter (BF).** We use the on-line mixture of Gaussians implementation of Seth Benton [3] with  $K = 2$  mixture components. The time constant for updating the distributions, see [11], is set to  $\alpha = 10^{-4}$  in both datasets. The background model is expected to capture at least 80% of the mixture model, i.e., we set  $T = 0.8$  in Eq. (4). Both parameters are set to ensure that standing persons are not easily integrated in the background model.

**Trajectory filter (TF).** We use Dirk-Jan Kroon's implementation [4] of the Frangi vessel filter [12] with default parameters and scale parameter set to 3.

**Particle filter (PF).** We use  $N=150$  particles, the variance for the position noise is set to  $\sigma_x^2=0.8$  and  $\sigma_y^2=0.2$  and the variance for the velocity noise is set to  $\sigma_{u,v}^2 = 0.1$ , all variances are set with respect to the grid in the image plane.

Person detection takes approximately 600 ms per frame, BF about 6 ms, TF about 14 ms and PF about 10 ms using a single core of a 3 GHz processor. The computational bottleneck is clearly the detection part, the proposed filter steps are essentially real-time.

### 3.2 Improving Object Detection

Our CCF approach combines the strength of background and appearance based approaches, whereas the individual ones are going to fail, as depicted in Fig. 5. For instance, pixel-wise color background subtraction will simply focus on all moving objects. Besides, occluded persons are hard to discern in the noisy foreground mask. Furthermore, the pre-trained detector ignores temporal information completely and a global threshold does not allow to detect occluded persons in the presence of difficult background. Our approach overcomes those limitations since the location specific detection confidence values are modeled temporally.

**Details.** Two typical detector confidence distributions and the fitted Gaussian mixture models are visualized in Fig. 6. At the position marked in green in Fig. 6(a), one Gaussian is considered to represent the background activities,

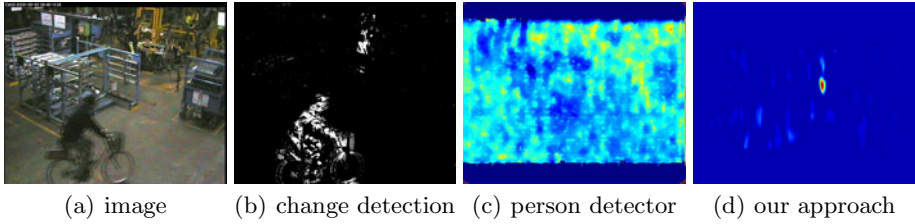
<sup>2</sup> Available at <http://sourceforge.net/projects/opencvlibrary>, 2010/02/24

<sup>3</sup> Available at

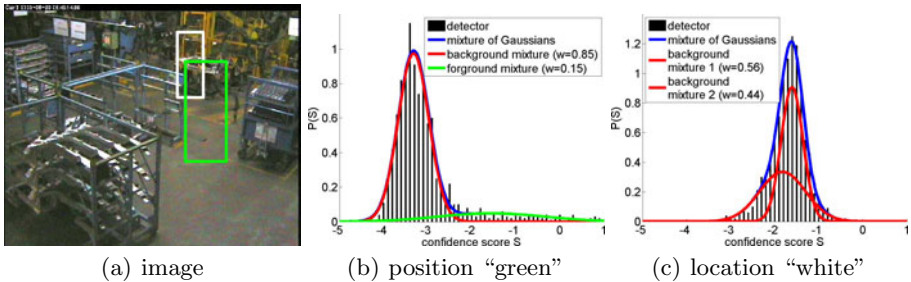
[http://www.sethbenton.com/mixture\\_of\\_gaussians.html](http://www.sethbenton.com/mixture_of_gaussians.html), 2010/03/07

<sup>4</sup> Available at

<http://www.mathworks.com/matlabcentral/fileexchange/24409-hessian-based-frangi-vesselness-filter>, 2010/02/24



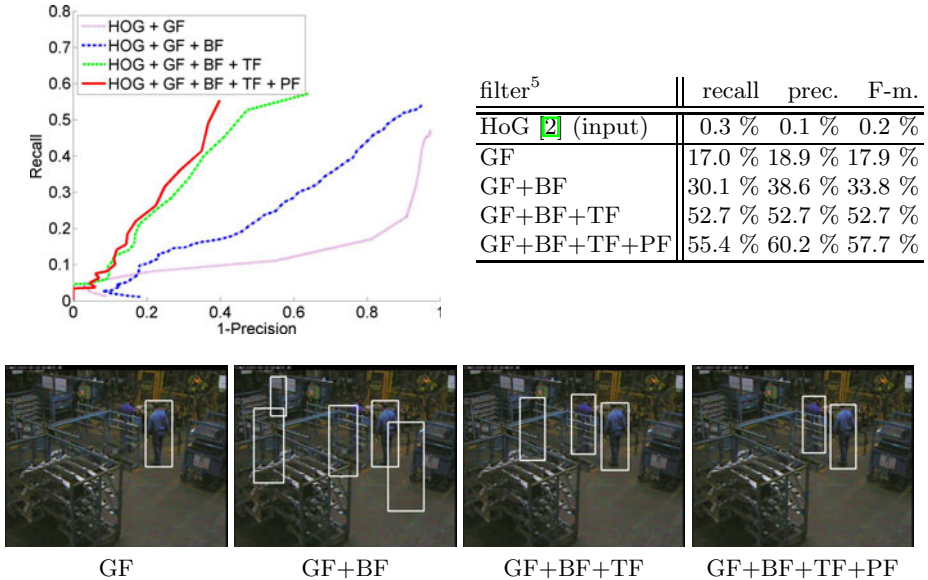
**Fig. 5.** Confidence scores of different methods. (b) simple background modeling with change detection also reports other moving objects while partially occluded persons are difficult to retrieve, (c) appearance based object detectors have difficulties in detecting occluded persons and discriminating the background, (d) our approach successfully refines the confidence map allowing for better detection results.



**Fig. 6.** Two typical mixture of Gaussians obtained by background modeling of [14] applied on detector confidence values. Mixture components which are considered to belong to the background model are shown in red, with  $T = 0.8$ .

whereas the second Gaussian (with a larger variance) is considered to model the activity of the object of interest at this location. At the second location depicted in 6(b), no persons are present (the welding machine is there) only parts of persons or sparks. Hence, both Gaussians with relatively high mean confidences belong to the background model. This also explains the drawbacks of using one global threshold for the whole image. Please note that these confidence statistics are computed over a relatively long time interval to not quickly integrate standing persons into the background model.

**Evaluation.** For a quantitative evaluation, we use recall-precision curves (RPCs). The recall corresponds to the detection rate whereas the precision relates to the trustfulness of a detection. In particular, a detection is accepted as true positive if it fulfills the overlap criterion of [14], i.e., a minimal overlap  $a_0 = (\text{area}(B_p) \cap \text{area}(B_{gt})) / (\text{area}(B_p) \cup \text{area}(B_{gt}))$  of 50 % is needed between the predicted bounding box  $B_p$  and the ground truth bounding box  $B_{gt}$  to count as true positive. Additionally, we also report recall and precision at maximized F-measure which is the harmonic mean between recall and precision.



**Fig. 7.** Recall-precision curves for each individual filter step in the SCOVIS dataset. Bottom: Example detections at maximized f-Measure

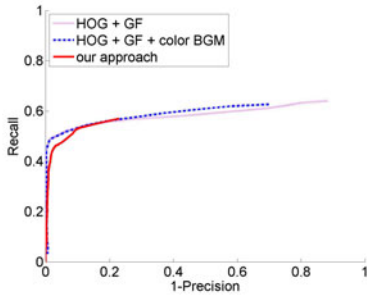
The RPC of each individual filter step in the SCOVIS dataset is depicted in Fig. 7 in conjunction with example detections. Improved results are manifested after each individual filter step of CCF in terms of increased recall and precision. Since the filtering steps are complementary the performance can be boosted by roughly 15-20 % in terms of f-Measure each. In particular, the geometric filter does not consider detections at inappropriate scales, the background filter adjust locally the detection thresholds (increasing the sensitivity) while the trajectory filter enforces temporal consistency. The post-processing to better align the detections increases precision and recall only slightly.

**Comparisons.** Quantitative and qualitative comparisons are shown in Fig. 8 and Fig. 9 for the i-LIDS AB Easy dataset and the challenging SCOVIS dataset, respectively. As baseline, we consider the detection results after the geometric filter since similar assumptions, e.g., about the ground plane, are often found in literature, e.g., [4]. The i-LIDS AB Easy datasets was chosen intentionally to illustrate the formidable performance of the person detector which is not lowered by CCF. In particular, the datasets contains several standing persons whose detection is shown to not be affected by the filter cascade.

However, in the more challenging SCOVIS dataset with poor detection results, CCF significantly improves the results by about 40 % in terms of f-Measure. Detailed results of applying our algorithm on both datasets are depicted in Fig. 10.

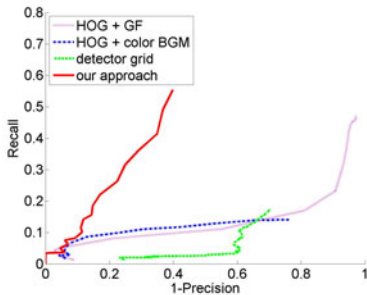
<sup>5</sup> HoG: Histogram of Gradients person detector [2]; GF: geometric filter; BF: background filter; TF: trajectory filter; PF: particle filter.



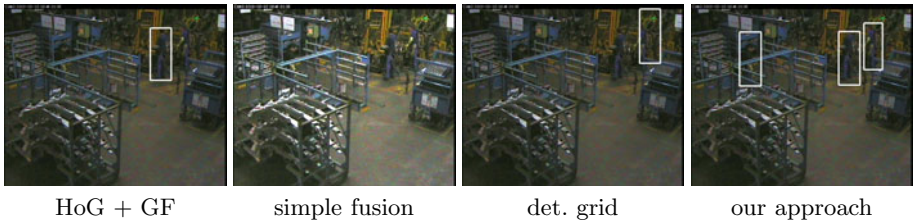


approach	recall	prec.	F-m.
HoG [2]+GF	54.7 %	86.6 %	67.0 %
simple fusion	54.3 %	87.5 %	67.0 %
our approach	53.3 %	89.4 %	66.8 %

**Fig. 8.** Results for the i-LIDS dataset in comparison to other approaches. Please note that the ground truth includes fully occluded persons which are impossible to detect. This said, the input is nearly perfect and all approaches perform similarly.



approach	recall	prec.	F-m.
HoG [2]+GF <sup>6</sup>	17.0 %	18.9 %	17.9 %
simple fusion	12.9 %	47.2 %	20.2 %
det. grid [6]	17.6 %	29.5 %	22.1 %
our approach	55.4 %	60.2 %	57.7 %



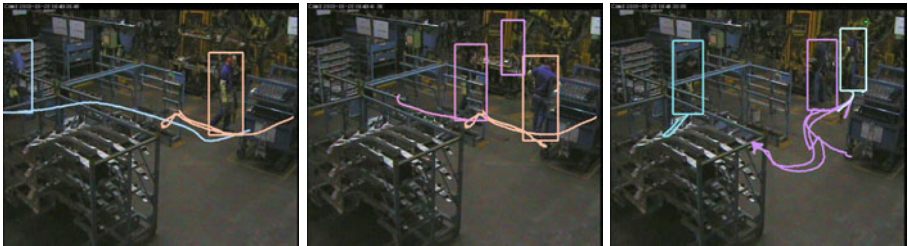
**Fig. 9.** Results for the SCOVIS dataset in comparison to other approaches

Additionally, we also evaluated a simple combination of traditional background modeling and human detection, i.e., a detection is only taken into account if at least 20 % of the bounding box is not modeled as foreground. However, this combination does not improve the results significantly as it is just a verification step a posteriori. We also compared our improved detections to the recently proposed approach of learned location specific classifiers [6]<sup>7</sup> which aims to include background information during learning. Whereas they are able to show improved results on common datasets, the results clearly shows that such an approach can not cope well with the challenges of the SCOVIS dataset.

<sup>6</sup> The part-based person detector of [3] achieved a recall of 2.3 % at a precision of 63.3 % with geometric filtering and default parameters.



**Fig. 10.** Short tracklets obtained by using the proposed CCF approach to improve on the HoG detection confidence. More examples are given on the authors' web page.



**Fig. 11.** Complete trajectories found by [9] using the improved detection results of our approach. The approach performs dismally if the HoG detections are directly passed to their detection association method.

### 3.3 Improving Tracking-by-Detection

Recent work explores tracking-by-detection [8,9], i.e. applying an object detector in each frame and then associating the detections across frames. The post-processing of CCF links the detections similarly, but at a lower level without extrapolation. To indicate improved tracking-by-detection results, we employ the approach of [9] which performs global trajectory association in an hierarchical manner<sup>7</sup>. The experiment was run twice, (i) providing the raw HoG detections and (ii) the improved ones obtained by CCF. Whereas the approach is performing dismally using the raw detections (not shown here), long trajectories are output when using CCF as pre-processing. Tracking results are depicted in Fig. 11.

<sup>7</sup> We gratefully thank the authors for applying their code on the SCOVIS dataset.

## 4 Conclusion and Future Work

We presented a novel method for filtering the detection confidences in surveillance scenarios. Our approach remains very general, only requiring a static camera and a sliding-windows object detector. CCF involves geometric constraints, long-term temporal constraints to suppress the background confidence distribution and short-term smoothness constraints on possible trajectories. The experimental evaluation on the task of person detection shows significant improvement over the input object detection results, especially in the case of occlusions and cluttered background. The approach does not learn specific object models, incorporate scene specific constraints, reason about complete trajectories, or use multiple cameras. All those extensions remain to be explored to further robustify tracking-by-detection methods.

**Acknowledgments.** This research was supported by the European Community's Seventh Framework Programme under grant agreement no FP7-ICT-216465 SCOVIS. We further thank Lee Middleton and Christine Tanner for inspiring discussions.

## References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR, vol. I, pp. 511–518 (2001)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR, vol. 1, pp. 886–893 (2005)
3. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: Proc. CVPR (2008)
4. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: Proc. CVPR., vol. 2, pp. 2137–2144 (2006)
5. Li, Y., Wu, B., Nevatia, R.: Human detection by searching in 3d space using camera and scene knowledge. In: Proc. ICPR (2008)
6. Roth, P., Sternig, S., Grabner, H., Bischof, H.: Classifier grids for robust adaptive object detection. In: Proc. CVPR (2009)
7. Stalder, S., Grabner, H., Gool, L.V.: Exploring context to learn scene specific object detectors. In: Proc. PETS (2009)
8. Leibe, B., Schindler, K., Gool, L.V.: Coupled detection and trajectory estimation for multi-object tracking. In: Proc. ICCV (2007)
9. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
10. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: Proc. ICCV (2009)
11. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. CVPR, vol. II, pp. 246–252 (1999)
12. Frangi, A., Niessen, W., Vincken, K., Viergever, M.: Multiscale vessel enhancement filtering, pp. 130–137 (1998)
13. Florin, C., Paragios, N., Williams, J.: Particle filters, a quasi-monte carlo solution for segmentation of coronaries. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 246–253. Springer, Heidelberg (2005)
14. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge (VOC 2009) Results (2009)

# Inter-camera Association of Multi-target Tracks by On-Line Learned Appearance Affinity Models

Cheng-Hao Kuo, Chang Huang, and Ram Nevatia

University of Southern California, Los Angeles, CA 90089, USA  
{chenghak,huangcha,nevatia}@usc.edu

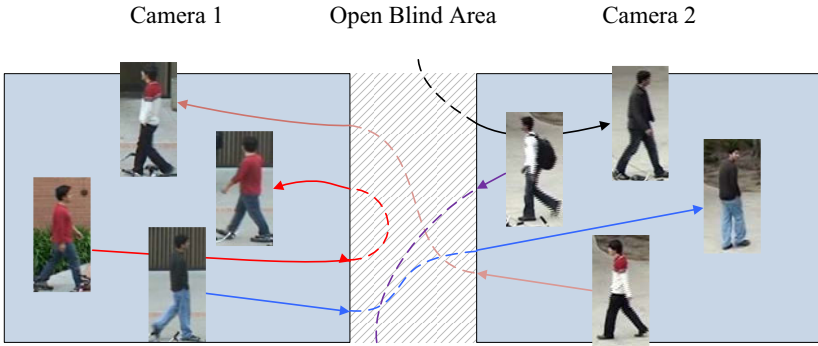
**Abstract.** We propose a novel system for associating multi-target tracks across multiple non-overlapping cameras by an on-line learned discriminative appearance affinity model. Collecting reliable training samples is a major challenge in on-line learning since supervised correspondence is not available at runtime. To alleviate the inevitable ambiguities in these samples, Multiple Instance Learning (MIL) is applied to learn an appearance affinity model which effectively combines three complementary image descriptors and their corresponding similarity measurements. Based on the spatial-temporal information and the proposed appearance affinity model, we present an improved inter-camera track association framework to solve the “target handover” problem across cameras. Our evaluations indicate that our method have higher discrimination between different targets than previous methods.

## 1 Introduction

Multi-target tracking is an important problem in computer vision, especially for applications such as visual surveillance systems. In many scenarios, multiple cameras are required to monitor a large area. The goal is to locate targets, track their trajectories, and maintain their identities when they travel within or across cameras. Such a system consists of two main parts: 1) intra-camera tracking, *i.e.* tracking multiple targets within a camera; 2) inter-camera association, *i.e.* “handover” of tracked targets from one camera to another. Although there have been significant improvements in intra-camera tracking, inter-camera track association when cameras have non-overlapping fields of views (FOVs) remains a less explored topic, which is the problem we focus on in this paper.

An illustration for inter-camera association of multiple tracks is shown in Figure 1. Compared to intra-camera tracking, inter-camera association is more challenging because 1) the appearance of a target in different cameras may not be consistent due to different sensor characteristics, lighting conditions, and viewpoints; 2) the spatio-temporal information of tracked objects between cameras becomes much less reliable. Besides, the open blind area significantly increases the complexity of the inter-camera track association problem.

Associating multiple tracks in different cameras can be formulated as a correspondence problem. Given the observations of tracked targets, the goal is to find the associated pairs of tracks which maximizes a joint linking probability,



**Fig. 1.** Illustration of inter-camera association between two non-overlapping cameras. Given tracked targets in each camera, our goal is to find the optimal correspondence between them, such that the associated pairs belong to the same object. A target may walk across the two cameras, return to the original one, or exit in the blind area. Also, a target entering Camera 2 from blind area is not necessarily from Camera 1, but may be from somewhere else. Such open blind areas significantly increase the difficulty of the inter-camera track association problem.

in which the key component is the affinity between tracks. For the affinity score, there are generally two main cues to be considered: the spatio-temporal information and appearance relationships between two non-overlapping cameras. Compared to spatial-temporal information, the appearance cues are more reliable for distinguishing different targets especially in cases where FOVs are disjoint. However, such cues are also more challenging to design since the appearances of targets are complex and dynamic in general. A robust appearance model should be adaptive to the current targets and environments.

A desired appearance model should incorporate discriminative properties between correct matches and wrong ones. Between a set of tracks among two non-overlapping cameras, the aim of the affinity model is to distinguish the tracks which belong to the same target from those which belong to different targets. Previous methods [1,2,3] mostly focused on learning the appearance models or mapping functions based on the correct matches, but no negative information is considered in their learning procedure. To the best of our knowledge, online learning of a discriminative appearance affinity model across cameras has not been utilized.

Collecting positive and negative training samples on-line is difficult since no hand-labelled correspondence is available at runtime. Hence, traditional learning algorithms may not apply. However, by observing spatio-temporal constraints of tracks between two cameras, some potentially associated pairs of tracks and some impossible pairs are formed as “weakly labelled samples”. We propose to adopt the Multiple Instance Learning (MIL) [4,5,6] to accommodate the ambiguity of labelling during the model learning process. Then the learned discriminative appearance affinity model is combined with spatio-temporal information to

compute the crucial affinities in the track association framework, achieving a robust inter-camera association system. It can be incorporated with any intra-camera tracking method to solve the problem of multi-object tracking across non-overlapping cameras.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. The overview of our approach is given in Section 3. The framework of track association between two cameras is described in Section 4. The method of learning a discriminative appearance affinity model using multiple instance learning is discussed in Section 5. The experimental results are shown in Section 6. The conclusion is given in Section 7.

## 2 Related Work

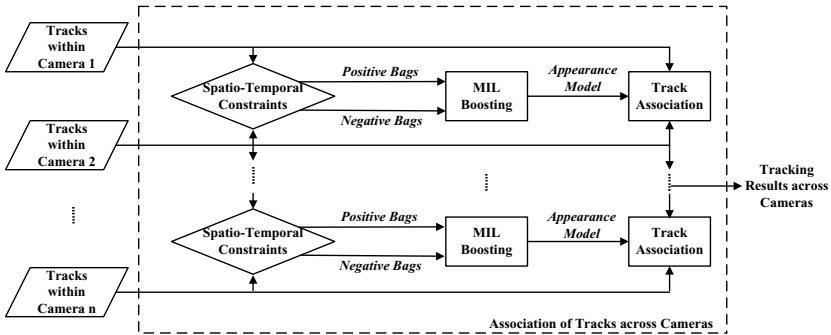
There is large amount of work, *e.g.* [7,8,9], for multi-camera tracking with overlapping field of views. These methods usually require camera calibration and environmental models to track targets. However, the assumption that cameras have overlapping fields of view is not always practical due to the large number of cameras required and the physical constraints upon their placement.

In the literature, [10,11,12] represent some early work for multi-camera tracking with non-overlapping field of views. To establish correspondence between objects in different cameras, the spatio-temporal information and appearance relationship are two important cues. For the spatio-temporal cue, Javed *et al.* [13] proposed a method to learn the camera topology and path probabilities of objects using Parzen windows. Dick and Brooks [14] used a stochastic transition matrix to describe people's observed patterns of motion both within and between fields of view. Makris *et al.* [15] investigated the unsupervised learning of a model of activity from a large set of observations without hand-labeled correspondence.

For the appearance cue, Porikli [1] derived a non-parametric function to model color distortion for pair-wise camera combinations using correlation matrix analysis and dynamic programming. Javed *et al.* [2] showed that the brightness transfer functions(BTFs) from a given camera to another camera lie in a low dimensional subspace and demonstrated that this subspace can be used to compute appearance similarity. Gilbert and Bowden [16] learned the BTFs incrementally based on Consensus-Color Conversion of Munsell color space [17].

Besides, there is some work addressing the optimization framework of multiple targets correspondence. Kettner and Zabih [12] used a Bayesian formulation to reconstruct the paths of targets across multiple cameras. Javed *et al.* [13] dealt with this problem by maximizing the a posteriori probability using a graph-theoretic framework. Song and Roy-Chowdhury [18] proposed a multi-objective optimization framework by combining short-term feature correspondences across the cameras with long-term feature dependency models.

Learning a discriminative appearance affinity model across non-overlapping cameras at runtime makes our approach different from the existing ones. Most previous methods did not incorporate any discriminative information to distinguish different targets, which is important for inter-camera track association especially when the scene contains multiple similar targets.



**Fig. 2.** The block diagram of our system for associating multiple tracked targets from multiple non-overlapping cameras

### 3 Overview of our Approach

Our system contains three main components: the method of collecting online training samples, the discriminative appearance affinity model, and track association framework. We use a time sliding window method to process video sequences. The learned appearance affinity models are updated in each time window. The system block diagram of our method is shown in Figure 2.

The collection of online training samples is obtained by observing the spatio-temporal constraints in a time sliding window. Assuming that the multi-object tracking is finished in each camera, a training sample is defined as a pair of tracks from two cameras respectively. Negative samples are collected by extracting pairs of tracks in two cameras which overlap in time. It is based on the assumption that one object can not appear in two non-overlapping cameras at the same time. Positive samples could be collected by similar spatio-temporal information. However, it is difficult to label the positive training sample in an online manner since it is indeed the correspondence problem that we want to solve. Instead of labelling each sample, several potentially linked pairs of tracks constitute one positive “bag”, which is suitable for the Multiple Instance Learning (MIL) algorithm.

The learning of appearance affinity model is to determine whether two tracks from different cameras belong to the same target or not according to their appearance descriptors and similarity measurements. Instead of using only color information as in previous work, appearance descriptors consisting of the color histogram, the covariance matrix, and the HOG feature, are computed at multiple locations to increase the power of description. Similarity measurements based on those features among the training samples establish the feature pool. Once the training samples are collected in a time sliding window, a MIL boosting algorithm is applied to select discriminative features from this pool and their corresponding weighted coefficients, and combines them into a strong classifier in the same time sliding window so that the learned models are adapted to the

current scenario. The prediction confidence output by this classifier is transformed to a probability space, which cooperates with other cues (e.g. spatial correspondence and time interval) to compute the affinity between tracks for association.

The association of tracks in two cameras is formulated as a standard assignment problem. A correspondence matrix is defined where the pairwise association probabilities are computed by spatio-temporal cues and appearance information. This matrix is designed to consider all possible scenarios in two non-overlapping cameras. The Hungarian algorithm is applied to solve this problem efficiently.

### 4 Track Association between Cameras

To perform track association across multiple cameras, we firstly focus on the track association between two cameras and then extend it to the case of multiple cameras. Previous methods often model it as an MAP problem to find the optimal solution via Bayes Theorem [12,3], a graph theoretic approach [13], and expected weighted similarity [19]. We present an efficient yet effective approach which maximizes the joint linking probability. Assuming that the task of single camera tracking has been already solved; there are  $m$  tracks in camera  $C^a$  denoted by  $\mathcal{T}^a = \{T_1^a, \dots, T_m^a\}$  and  $n$  tracks in camera  $C^b$  denoted by  $\mathcal{T}^b = \{T_1^b, \dots, T_n^b\}$  respectively. We may simply create a  $m$  by  $n$  matrix and find the optimal correspondence between  $\mathcal{T}^a$  and  $\mathcal{T}^b$ . However, in the case of non-overlapping cameras, there exist “blind” areas where objects are invisible. For example, an object which leaves  $C^a$  does not necessarily enter  $C^b$  as it may either go to the exit in the blind area or return to  $C^a$ . We define an extended correspondence matrix of size  $(2m + 2n) \times (2m + 2n)$  as follows:

$$\mathbf{H} = \left[ \begin{array}{cc|cc} \mathbf{A}_{m \times m} & \mathbf{B}_{m \times n} & \mathbf{F}_{m \times m} & -\infty_{m \times n} \\ \mathbf{D}_{n \times m} & \mathbf{E}_{n \times n} & -\infty_{n \times m} & \mathbf{G}_{n \times n} \\ \hline \mathbf{J}_{m \times m} & -\infty_{m \times n} & & \\ -\infty_{n \times m} & \mathbf{K}_{n \times n} & & \mathbf{0}_{(m+n) \times (m+n)} \end{array} \right] \tag{1}$$

This formulation is inspired by [20], but we made the necessary modification to accommodate all situation which could happen between the tracks of two non-overlapping cameras. The components of each matrix are defined as follows:  $B_{ij} = \log P_{link}(T_i^a \rightarrow T_j^b)$  is the linking score of that the tail of  $T_i^a$  links to the head of  $T_j^b$ . It models the situation that a target leaves  $C^a$  and then enters  $C^b$ ; a similar description is applied to  $D_{ij} = \log P_{link}(T_j^a \rightarrow T_i^b)$ .  $A_{ij} = \log P_{link}(T_i^a \rightarrow T_j^a)$  if  $i \neq j$  is the linking score of that the tail of  $T_i^a$  links to the head of  $T_j^a$ . It models the situation that a target leaves  $C^a$  and then re-enters camera  $a$  without travelling to camera  $C^b$ ; a similar description is also applied to  $E_{ij} = \log P_{link}(T_i^b \rightarrow T_j^b)$  if  $i \neq j$ .  $F_{ij}$  or  $G_{ij}$  if  $i = j$  is the score of the  $T_i^a$  or  $T_j^b$  is terminated. It models the situation that the head of target can not be linked to the tail of any tracks.  $J_{ij}$  and  $K_{ij}$  if  $i = j$  is the score of that the  $T_i^a$  or  $T_j^b$  is initialized. It models the situation that the tail of target can not link to the head of any track. By applying the Hungarian algorithm to  $\mathbf{H}$ , the optimal



**Table 1.** A short summary of the elements in each sub-matrix in  $\mathbf{H}$ , which models all possible situations between the tracks of two non-overlapping cameras. The optimal assignment is solved by Hungarian algorithm.

matrix	description	element
<b>A</b>	the target leaves and returns to $C^a$	$A_{ij} = -\infty$ if $i = j$
<b>B</b>	the target leaves $C^a$ and enters $C^b$	$B_{ij}$ is a full matrix
<b>D</b>	the target leaves $C^b$ and enters $C^a$	$D_{ij}$ is a full matrix
<b>E</b>	the target leaves and returns to $C^b$	$E_{ij} = -\infty$ if $i = j$
<b>F</b>	the target terminates in $C^a$	$F_{ij} = -\infty$ if $i \neq j$
<b>G</b>	the target terminates in $C^b$	$G_{ij} = -\infty$ if $i \neq j$
<b>J</b>	the target is initialized in $C^a$	$J_{ij} = -\infty$ if $i \neq j$
<b>K</b>	the target is initialized in $C^b$	$K_{ij} = -\infty$ if $i \neq j$

assignment of association is obtained efficiently. A summary of each sub-matrix in  $\mathbf{H}$  is given in Table 1.

The linking probability, *i.e.* affinity between two tracks  $T_i$  and  $T_j$  is defined as the product of three important cues (appearance, space, time):

$$P_{link}(T_i \rightarrow T_j) = P_a(T_i, T_j) \cdot P_s(e(T_i), e(T_j)) \cdot P_t(T_i \rightarrow T_j | e(T_i), e(T_j)) \quad (2)$$

where  $e(T_i)$  denotes the exit/entry region of  $T_i$ . Each of three components measures the likelihood of  $T_i$  and  $T_j$  being the same object. The latter two terms  $P_s$  and  $P_t$  are spatio-temporal information which can be learned automatically by the methods proposed in [15, 3]. We focus on the first term  $P_a$  and propose a novel framework of online learning a discriminative appearance affinity model.

## 5 Discriminative Appearance Affinity Models with Multiple Instance Learning

Our goal is to learn a discriminative appearance affinity model across the cameras at runtime. However, how to choose positive and negative training samples is a major challenge since exact hand-labelled correspondence is not available while learning online. Based on the spatio-temporal constraints, we are able to only exclude some impossible links and retain several possible links, which are called “weakly labelled training examples”.

Recent work [5, 6] presents promising results on face detection and visual tracking respectively using Multiple Instance Learning (MIL). Compared to traditional discriminative learning, MIL describes that samples are presented in “bags”, and the labels are provided for the bags instead of individual samples. A positive “bag” means it contains at least one positive sample; a negative bag means all samples in this bag are negative. Since some flexibility is allowed for the labelling process, we may use the “weakly labelled training examples” by spatio-temporal constraints and apply a MIL boosting algorithm to learn the discriminative appearance affinity model.

## 5.1 Collecting Training Samples

We propose a method to collect weakly labelled training samples using spatio-temporal constraints. To learn an appearance affinity model between cameras, a training sample is defined as a pair of tracks from two cameras respectively. Based on the tracks generated by a robust single camera multi-target tracker, we make a conservative assumption: any two tracks from two non-overlapping cameras which overlap in time represent different targets. It is based on the observation that one target can not appear at different locations at the same time. Positive samples are more difficult to obtain since there is no supervised information to indicate which two tracks among two cameras represent the same objects. In other words, the label of “+1” can not be assigned to individual training samples. To deal with the challenging on-line labelling problem, we collect possible pairs of tracks by examining spatio-temporal constraints and put them into a “bag” which is labelled “+1”. The MIL boosting is applied to learn the desired discriminative appearance affinity model.

In our implementation, there are two set to be formed for each track: a set of “similar” tracks and a set of “discriminative” tracks. For a certain track  $T_j^a$  in camera  $C^a$ , each element in its “discriminative” set  $\mathcal{D}_j^b$  indicates a target  $T_k^b$  in camera  $C^b$  which is impossible to be the same target with  $T_j^a$ ; each element in the “similar” set  $\mathcal{S}_j^b$  represents a possible target  $T_k^b$  in  $C^b$  which might be the same target with  $T_j^a$ . These cases are described as:

$$\begin{aligned} T_k^b \in \mathcal{S}_j^b & \text{ if } P_s(T_j^a \rightarrow T_k^b) \cdot P_t(e(T_j^a), e(T_k^b)) > \theta \\ T_k^b \in \mathcal{D}_j^b & \text{ if } P_s(T_j^a \rightarrow T_k^b) \cdot P_t(e(T_j^a), e(T_k^b)) = 0 \end{aligned} \quad (3)$$

The threshold  $\theta$  is adaptively chosen to maintain a moderate number of instances included in each positive bag. The training sample set  $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$  can be denoted by

$$\begin{aligned} \mathcal{B}^+ &= \left\{ x_i : \{T_j^a, T_k^b\}, \forall T_k^b \in \mathcal{S}_j^b; y_i : +1 \right\} \\ \mathcal{B}^- &= \left\{ x_i : (T_j^a, T_k^b), \text{ if } T_k^b \in \mathcal{D}_j^b; y_i : -1 \right\} \end{aligned} \quad (4)$$

where each training sample  $x_i$  may contain multiple pairs of tracks which represents a bag. A label is given to a bag.

## 5.2 Representation of Appearance Model and Similarity Measurement

To build a strong appearance model, we begin by computing several local features to describe a tracked target. In our design, three complementary features: color histograms, covariance matrices, and histogram of gradients (HOG) constitute the feature pool. Given a tracked target, features are extracted at different locations and different scales from the head and tail part to increase the descriptive ability.

We use RGB color histograms to represent the color appearance of a local image patch. Histograms have the advantage of being easy to implement and having well studied similarity measures. Single channel histograms are concatenated to form a vector  $\mathbf{f}_{RGB_i}$ , but any other suitable color space can be used. In our implementation, we use 8 bins for each channel to form a 24-element vector. To describe the image texture, we use a descriptor based on covariance matrices of image features proposed in [21]. It has been shown to give good performance for texture classification and object categorization. To capture shape information, we choose the Histogram of Gradients (HOG) Feature proposed in [22]. In our design, a 32D HOG feature  $\mathbf{f}_{HOG_i}$  is extracted over the region  $R$ ; it is formed by concatenating 8 orientations bins in  $2 \times 2$  cells over  $R$ .

In summary, the appearance descriptor of a track  $T_i$  can be written as:

$$\mathcal{A}_i = (\{\mathbf{f}_{RGB_i}^l\}, \{\mathbf{C}_i^l\}, \{\mathbf{f}_{HOG_i}^l\}) \quad (5)$$

where  $\mathbf{f}_{RGB_i}^l$  is the feature vector for color histogram,  $\mathbf{C}_i^l$  is the covariance matrix, and  $\mathbf{f}_{HOG_i}^l$  is the 32D HOG feature vector. The superscript  $l$  means that the features are evaluated over region  $R^l$ .

Given the appearance descriptors, we can compute similarity between two patches. The color histogram and HOG feature are histogram-based features so standard measurements, such as  $\chi^2$  distance, Bhattacharyya distance, and correlation coefficient can be used. In our implementation, correlation coefficient is chosen for simplicity. The distance measurement of covariance matrices is determined by solving a generalized eigenvalues problem, which is described in [21].

After computing the appearance model and the similarity between appearance descriptors at different regions, we form a feature vector by concatenating the similarity measurements with different appearance descriptors at multiple locations. This feature vector gives us a feature pool that we can use an appropriate boosting algorithm to construct a strong classifier.

### 5.3 Multiple Instance Learning

Our goal is to design a discriminative appearance model which determines the affinity score of appearance between two objects in two different cameras. Again, a sample is defined as a pair of targets from two cameras respectively. The affinity model takes a pair of objects as input and returns a score of real value by a linear combination of weak classifiers. The larger the affinity score, the more likely that two objects in one sample represent the same target. We adopt the MIL Boosting framework proposed in [5] to select the weak classifiers and their corresponding weighted coefficients. Compared to conventional discriminative boosting learning, training samples are not labelled individually in MIL; they form “bags” and the label is given to each bag, not to each sample. Each sample is denoted by  $x_{ij}$ , where  $i$  is the index for the bag and  $j$  is the index for the sample within the bag. The label of each bag is represented by  $y_i$  where  $y_i \in \{0, 1\}$ .

Although the known labels are given to bags instead of samples, the goal is to learn the the instance classifier which takes the following form:

$$H(x_{ij}) = \sum_{t=1}^T \alpha_t h_t(x_{ij}) \tag{6}$$

In our framework, the weak hypothesis is from the feature pool obtained by Section 5.2. We adjust the sign and normalize  $h(x)$  to be in the restricted range  $[-1, +1]$ . The sign of  $h(x)$  is interpreted as the predicted label and the magnitude  $|h(x)|$  as the confidence in this prediction.

The probability of a sample  $x_{ij}$  being positive is defined as the standard logistic function,

$$p_{ij} = \sigma(y_{ij}) = \frac{1}{1 + \exp(-y_{ij})} \tag{7}$$

where  $y_{ij} = H(x_{ij})$ . The probability of a bag being positive is defined by the “noisy OR” model:

$$p_i = 1 - \prod_j (1 - p_{ij}) \tag{8}$$

If one of the samples in a bag has a high probability  $p_{ij}$ , the bag probability  $p_i$  will be high as well. This property is appropriate to model that a bag is labelled as positive if there is at least one positive sample in this bag. MIL boosting uses the gradient boosting framework to train a boosting classifier that maximizes the log likelihood of bags:

$$\log L(H) = \sum_i y_i \log p_i + (1 - y_i) \log(1 - p_i) \tag{9}$$

The weight of each sample is given as the derivative of the loss function  $\log L(H)$  with respect to the score of that sample  $y_{ij}$ :

$$w_{ij} = \frac{\partial \log L(H)}{\partial y_{ij}} = \frac{y_i - p_i}{p_i} p_{ij} \tag{10}$$

Our goal is to find  $H(x)$  which maximizes (9), where  $H(x)$  can be obtained by sequentially adding new weak classifiers. In the  $t$ -th boosting round, we aim at learning the optimal weak classifier  $h_t$  and weighted coefficient  $\alpha_t$  to optimize the loss function:

$$(\alpha_t, h_t) = \arg \min_{h, \alpha} \log L(H_{t-1} + \alpha h) \tag{11}$$

To find to the optimal  $(\alpha_t, h_t)$ , we follow the framework used in [23, 5] which views boosting as a gradient descent process, each round it searches for a weak classifier  $h_t$  to maximize the gradient of the loss function. Then the weighted coefficient  $\alpha_t$  is determined by a linear search to maximize  $\log L(H + \alpha_t h_t)$ . The learning procedure is summarized in Algorithm 1.

---

**Algorithm 1.** Multiple Instance Learning Boosting

---

$\mathcal{B}^+ = \left\{ \left( \{x_{i1}, x_{i2}, \dots\}, +1 \right) \right\}$ : Positive bags  
**Input:**  $\mathcal{B}^- = \left\{ \left( \{x_{i1}, x_{i2}, \dots\}, -1 \right) \right\}$ : Negative bags  
 $\mathcal{F} = \{ \mathbf{h}(x_{ij}) \}$ : Feature pools

- 1: Initialize  $H = 0$
- 2: **for**  $t = 1$  to  $T$  **do**
- 3:   **for**  $k = 1$  to  $K$  **do**
- 4:      $p_{ij}^k = \sigma(H + h_k(x_{ij}))$
- 5:      $p_i^k = 1 - \prod_j (1 - p_{ij}^k)$
- 6:      $w_{ij}^k = \frac{y_i - p_i^k}{p_i^k} p_{ij}^k$
- 7:   **end for**
- 8:   Choose  $k^* = \arg \max_k \sum_{ij} w_{ij}^k h_k(x_{ij})$
- 9:   Set  $h_t = h_{k^*}$
- 10:   Find  $\alpha^* = \arg \max_{\alpha} \log L(H + \alpha h_t)$  by linear search
- 11:   Set  $\alpha_t = \alpha^*$
- 12:   Update  $H \leftarrow H + \alpha_t h_t$
- 13: **end for**

**Output:**  $H(x) = \sum_{t=1}^T \alpha_t h_t(x)$

---

## 6 Experimental Results

The experiments are conducted on a three-camera setting with disjoint FOVs. First, we evaluate the effectiveness of our proposed on-line learned discriminative appearance affinity model by formulating the correspondence problem as a binary classification problem. Second, for a real scenario of multiple non-overlapping cameras, the evaluation metric is defined, and the tracking results using our proposed system are presented. It is shown that our method achieves good performance in a crowded scene. Some graphical examples are also provided.

### 6.1 Comparison of Discriminative Power

We first evaluate the discriminative ability of our appearance affinity model, independent of the tracking framework that it will be embedded in. Given the tracks in each camera, we manually label the pairs of tracks that should be associated to from the ground truth. Affinity scores are computed among every possible pair in a time sliding window by four methods: (1) the correlation coefficients of two color histogram; (2) the model proposed in Section 5 but without MIL learning, *i.e.* with equal coefficients  $\alpha_t$ ; (3) off-line MIL learning, *i.e.* learning is done on another time sliding window; (4) MIL learning on the same time

**Table 2.** The comparison of the Equal Error Rate using different appearance affinity models. It shows that the on-line learning method has the most discriminative power.

Camera pair	color only	no learning	off-line learning	on-line learning
$C^1, C^2$	0.231	0.156	0.137	<b>0.094</b>
$C^2, C^3$	0.381	0.222	0.217	<b>0.159</b>

window. In a three-camera setting, the experiments are done in two camera pairs ( $C^1, C^2$ ) and ( $C^2, C^3$ ); equal error rate in two tasks is the metric to evaluate the performance. In ( $C^1, C^2$ ), the number of evaluated pairs is 434 and the number of positive pairs is 35. In ( $C^2, C^3$ ), the number of evaluated pairs is 148 and the number of positive pairs is 18. The length of time sliding window is 5000. The experimental results are shown in Table 2. In each camera pair, the model using online MIL learning achieves the lowest equal error rate compared to the other three methods.

## 6.2 Evaluation Metrics

In previous work, quantitative evaluation of multi-target tracking across multiple cameras is barely mentioned or simply a single number *e.g.* tracking accuracy is used. It is defined as the ratio of the number of objects tracked correctly to the total number of objects that passed through the scene in [23]. However, it may not be a suitable metric to measure the performance of a system fairly, especially in a crowded scene where targets have complicated interactions. For example, if two tracked targets exchange their identities twice while travelling across a series of three cameras should be worse than if they exchange only once. Nevertheless, these two situations are both counted as incorrect tracked objects in the metric of “tracking accuracy”. We need a more complete metric to evaluate the performance of inter-camera track association.

In the case of tracking within a single camera, fragments and ID switches are two commonly used metrics. We adopt the definitions used in [24] and apply it to the case of tracking across cameras. Assuming that multiple targets tracking in a single camera is obtained, we only focus on the fragments and ID switches which are not defined within cameras. Given the tracks in two cameras  $C^a$  and  $C^b$ :  $\mathcal{T}^a = \{T_1^a, \dots, T_m^a\}$  and  $\mathcal{T}^b = \{T_1^b, \dots, T_n^b\}$ , the metrics in tracking evaluation are:

- Crossing Fragments(X-Frag): The total number of times that there is a link between  $T_i^a$  and  $T_j^b$  in the ground truth, but missing in the tracking result.
- Crossing ID switches(X-IDS): The total number of times that there is no link between  $T_i^a$  and  $T_j^b$  in the ground truth, but existing in the tracking result.
- Returning Fragments(R-Frag): The total number of times that there is link between  $T_i^a$  and  $T_j^a$  which represents a target leaving and returning to  $C^a$  in ground truth, but missing in the tracking result.

**Table 3.** Tracking results using different appearance models with our proposed metrics. The lower the numbers, the better performance it is. It shows that our on-line learned appearance affinity models achieve the best results.

Method	X-Frag	X-IDS	R-Frag	R-IDS
(a)input tracks	206	0	15	0
(b)color only	9	18	12	8
(c)off-line learning	6	15	11	7
(d)on-line learning	<b>4</b>	<b>12</b>	<b>10</b>	<b>6</b>

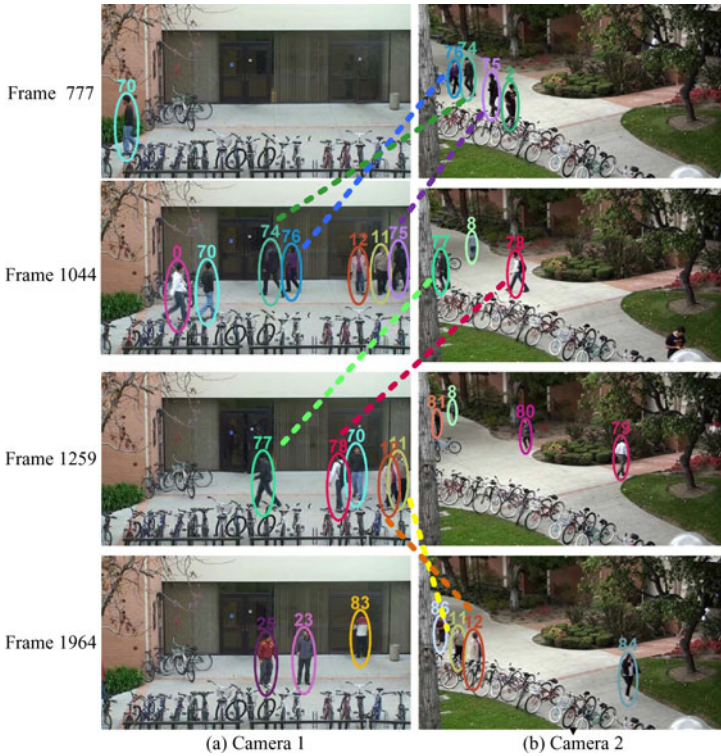
- Returning ID switches(R-IDS): The total number of times that there is no link between  $T_i^a$  and  $T_j^a$  which means they represent different targets in ground truth, but existing in the tracking result.

For example, there are  $T_1^a, T_2^a$  in  $C^a$ , and  $T_1^b, T_2^b$  in  $C^b$ . In the ground truth,  $(T_1^a, T_1^b)$  and  $(T_2^a, T_2^b)$  are the linked pairs. If they switch their identities in the tracking result, *i.e.*  $(T_1^a, T_2^b)$  and  $(T_2^a, T_1^b)$  are the linked pairs, that is considered as 2 X-frag and 2 X-IDS. This metric is more strict but well-defined than the traditional definition of fragments and ID switches. Similar descriptions apply to R-Frag and R-IDS. The lower these four metrics, the better is the tracking performance.

### 6.3 Tracking Results

The videos used in our evaluation are captured by three cameras in a campus environment with frame size of  $852 \times 480$  and length of 25 minutes. It is more challenging than the dataset used in the previous works in the literature since this dataset features a more crowded scene (2 to 10 people per frame in each camera). There are many inter-object occlusions and interactions and people walking across cameras occurs often. The multi-target tracker within a camera we use is based on [24], which is a detection-based tracking algorithm with hierarchical association.

We compare our approach with different appearance models. The results are also shown in Table 3. The result of (a) represents the input, *i.e.* no linking between any tracks in each camera. The result of (b) uses only color histogram is used as the appearance model. In the result of (c), our proposed appearance model is used but learned in an off-line environment, which means the coefficients  $\alpha_t$  are fixed. The result of (d) uses our proposed appearance models. It shows that our proposed on-line learning method outperforms these two appearance models. This comparison justifies that our stronger appearance model with on-line learning improves the tracking performance. Some association results are shown in Figure 3. It shows that our method finds the correct association among multiple targets in a complex scenen, *e.g.* people with IDs of 74, 75, and 76 when they travel from camera 2 to camera 1.



**Fig. 3.** Sample tracking results on our dataset. Some tracked people travelling through the cameras are linked by dotted lines. For example, the targets with IDs of 74, 75, and 76 leave Camera 2 around the same time, our method finds the correct association when they enter Camera 1. This figure is best viewed in color.

## 7 Conclusion

We describe a novel system for associating multi-target tracks across multiple non-overlapping cameras. The contribution of this paper focuses on learning a discriminative appearance affinity model at runtime. To solve the ambiguous labelling problem, we adopt Multiple Instance Learning boosting algorithm to learn the desired discriminative appearance models. An effective multi-object correspondence optimization framework for intra-camera track association problem is also presented. Experimental results on a challenging dataset show the robust performance by our proposed system.

**Acknowledgments.** This paper is based upon work supported in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract number W911NF-08-C-0068 and, in part, by Office of Naval Research under grant number N00014-10-1-0517.



## References

1. Porikli, F.: Inter-camera color calibration by correlation model function. In: ICIP (2003)
2. Javed, O., Shafique, K., Shah, M.: Appearance modeling for tracking in multiple non-overlapping cameras. In: CVPR (2005)
3. Chen, K.W., Lai, C.C., Hung, Y.P., Chen, C.S.: An adaptive learning method for target tracking across multiple cameras. In: CVPR (2008)
4. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T., Pharmaceutical, A.: Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 31–71 (1997)
5. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. In: NIPS (2005)
6. Babenko, B., Yang, M.H., Belongie, S.: Visual Tracking with Online Multiple Instance Learning. In: CVPR (2009)
7. Cai, Q., Aggarwal, J.: Tracking human motion in structured environments using a distributed-camera system. *IEEE Tran. on PAMI* 21, 1241–1247 (1999)
8. Collins, R., Lipton, A., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multi-sensor surveillance. *Proceedings of the IEEE* 89, 1456–1477 (2001)
9. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Tran. on PAMI* 25, 1355–1360 (2003)
10. Huang, T., Russell, S.: Object identification in a bayesian context. In: IJCAI (1997)
11. Pasula, H., Russell, S., Ostl, M., Ritov, Y.: Tracking many objects with many sensors. In: IJCAI (1999)
12. Kettner, V., Zabih, R.: Bayesian multi-camera surveillance. In: CVPR (1999)
13. Javed, O., Rasheed, Z., Shafique, K., Shah, M.: Tracking across multiple cameras with disjoint views. In: ICCV (2003)
14. Dick, A.R., Brooks, M.J.: A stochastic approach to tracking objects across multiple cameras. In: Australian Conference on Artificial Intelligence (2004)
15. Makris, D., Ellis, T., Black, J.: Bridging the gaps between cameras. In: CVPR (2004)
16. Gilbert, A., Bowden, R.: Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 125–136. Springer, Heidelberg (2006)
17. Sturges, J., Whitfield, T.: Locating basic colour in the munsell space. *Color Research and Application* 20, 364–376 (1995)
18. Song, B., Roy-Chowdhury, A.: Robust tracking in a camera network: A multi-objective optimization framework. *IEEE Journal of Selected Topics in Signal Processing* 2, 582–596 (2008)
19. Song, B., Roy-Chowdhury, A.K.: Stochastic adaptive tracking in a camera network. In: ICCV (2007)
20. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
21. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
23. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent in function space (1999)
24. Kuo, C.H., Huang, C., Nevatia, R.: Multi-target tracking by on-line learned discriminative appearance models. In: CVPR (2010)

# Multi-person Tracking with Sparse Detection and Continuous Segmentation

Dennis Mitzel<sup>1</sup>, Esther Horbert<sup>1</sup>, Andreas Ess<sup>2</sup>, and Bastian Leibe<sup>1</sup>

<sup>1</sup> UMIC Research Centre RWTH Aachen University, Germany

<sup>2</sup> Computer Vision Laboratory, ETH Zurich, Switzerland

**Abstract.** This paper presents an integrated framework for mobile street-level tracking of multiple persons. In contrast to classic tracking-by-detection approaches, our framework employs an efficient level-set tracker in order to follow individual pedestrians over time. This low-level tracker is initialized and periodically updated by a pedestrian detector and is kept robust through a series of consistency checks. In order to cope with drift and to bridge occlusions, the resulting tracklet outputs are fed to a high-level multi-hypothesis tracker, which performs longer-term data association. This design has the advantage of simplifying short-term data association, resulting in higher-quality tracks that can be maintained even in situations where the pedestrian detector does no longer yield good detections. In addition, it requires the pedestrian detector to be active only part of the time, resulting in computational savings. We quantitatively evaluate our approach on several challenging sequences and show that it achieves state-of-the-art performance.

## 1 Introduction

In this paper, we address the problem of multi-person tracking with a camera mounted on top of a moving vehicle, *e.g.* a mobile robot. This task is very challenging, since multiple persons may appear or emerge from occlusions at every frame and need to be detected. Since background modeling [1] is no longer applicable in a mobile scenario, this is typically done using visual object detectors [2]. Consequently, tracking-by-detection has become the dominant paradigm for such applications [3–8]. In this framework, a generic person detector is applied to every frame of the input video sequence, and the resulting detections are associated to tracks. This leads to challenging data association problems, since the detections may themselves be noisy, containing false positives and misaligned detection bounding boxes [2]. Several approaches have been proposed to address this issue by optimizing over a larger temporal window using model selection [5], network flow optimization [9], or hierarchical [8] or MCMC data association [10].

Intuitively, this complex data association seems to be at least to some degree an overkill. Once we have detected a person in one frame, we know its appearance and should be able to use this information in order to disambiguate future data associations. This has been attempted by using person-specific color descriptors (*e.g.* [4–6]) or online-trained classifiers [11]. The difficulty here is however that no precise segmentation is given – the detector bounding boxes contain many background pixels and the persons’ limbs may undergo considerable articulations, causing the classifiers to drift.

Another problem of tracking systems that only rely on detector input is that they will not work in situations where the detectors themselves fail, *e.g.* when a person gets too close to the camera and is partially occluded by the image borders. [6] explicitly point out those situations as failure cases of their approach.

In this paper, we propose to address those problems by complementing the detection-based tracking framework with a robust image-level tracker based on level-set (LS) segmentation. In this integration, a high-level tracker only initializes new tracklets from object detections, while the frame-to-frame target following and data association is taken over by the image-based tracker. The resulting tracked target locations are then transmitted back to the high-level tracker, where they are integrated into 3D trajectories using physically plausible motion models.

This combination is made possible by the great progress LS segmentation and tracking approaches have made in recent years [12]. Approaches are now available that can obtain robust tracking performance over long and challenging sequences [13]. In addition, LS trackers can be efficiently implemented using narrow-band techniques, since they need to process only a small part of the image around the tracked contour. However, the targeted integration is far from trivial. The LS tracking framework has originally been developed for following individual targets over time and has mostly been evaluated for tasks where a manual initialization is given [12, 13]. Here, we need to automatically initialize a large number of tracklets from potentially inaccurate detections. In addition, we need to deal with overlaps and partial occlusions between multiple followed persons, as well as with tracker drift from changing lighting conditions and poor image contrast. Finally, we need to account for cases where a person gets fully occluded for a certain time and comes into view again a few frames later. In this paper, we show how those challenges can be addressed by a careful interplay of the system components.

Our paper makes the following contributions: (1) We demonstrate how LS trackers can be integrated into a tracking-by-detection framework for robust multi-person tracking. (2) Our approach is based on the idea to track each individual pedestrian by an automatically initialized level-set. We develop robust methods for performing this initialization from object detections and show how additional geometric constraints and consistency checks can be integrated into the image-based LS tracker. (3) The tracked person contours in each video frame are automatically converted to 3D world coordinates and are transmitted to the high-level tracker, which integrates the position evidence into a robust multi-hypothesis trajectory estimation approach making use of physical motion models. This high-level tracker is responsible for initializing new tracks, correcting the low-level tracker's predictions when drift occurs, and tracking person identities through occlusions. (4) We experimentally demonstrate that this proposed integration achieves robust multi-person tracking performance in challenging mobile scenarios. In particular, as our approach does not depend on continuous pedestrian detection, it can also continue tracking persons that are only partially visible. (5) An interesting property of our integration is that it does not require the object detector to be executed for every video frame. This is especially relevant for the deployment on mobile platforms, where real-time performance is crucial and computational resources are notoriously limited. We experimentally investigate at what intervals object detections are still required for robust system-level performance.

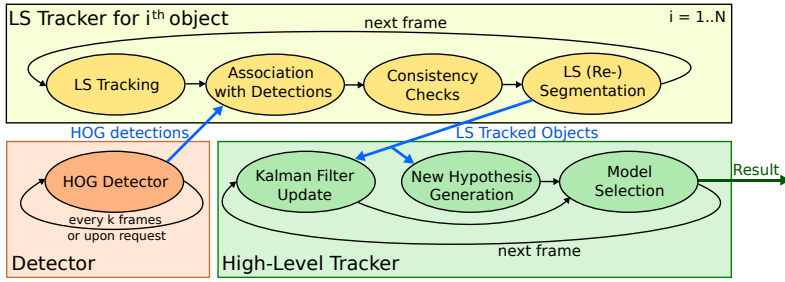


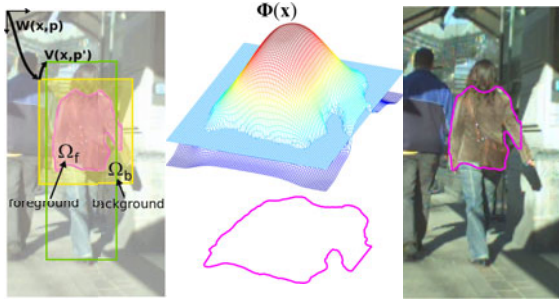
Fig. 1. System-level view of our proposed end-to-end tracking framework

The following section discusses related work. After that, Sec. 2 presents our proposed end-to-end tracking framework. Sec. 3 introduces the basic algorithmic components for LS tracking and trajectory estimation. Sec. 4 then describes the details of the integration, and Sec. 5 presents experimental results.

**Related Work.** Multi-object tracking from a mobile platform is a core capability for many applications in mobile robotics and autonomous vehicles [14]. While early approaches have been developed for aerial scenarios [15, 16], an application on ground-level poses significant additional difficulties. Robust multi-person tracking in such challenging situations has only recently become feasible by the development of powerful tracking-by-detection approaches [4–7, 14]. Various strategies have been developed for solving the challenging data association problems encountered here. However, most of them regard only a single-layer tracker [3, 5–7, 11], which sometimes makes the problem unnecessarily hard. Most directly related to our approach are the multi-layer models of [15, 16], which also initialize a number of low-level trackers to follow individual objects and later integrate their results in a high-level tracker. However, their frameworks are based on aerial scenarios, where adaptive background modeling is still feasible. [8] also propose a hierarchical data association framework that links detection responses to form tracklets at an image level, before fusing the tracklets and integrating scene constraints at higher levels. Their approach is however targeted at a surveillance application with a static camera. [17] integrates multiple short and low-confidence tracklet hypotheses into consistent tracks using MCMC. In contrast, our approach creates long and highly confident tracklets for individual persons under specific conditions of an LS tracker and integrates them into an EKF-based multiple-hypothesis tracker. To our knowledge, ours is the first approach that integrates segmentation-based LS-trackers [12, 13] with a tracking-by-detection framework for street-level mobile tracking.

## 2 Integrated Tracking Framework

Fig. 1 shows a system-level overview of our proposed integrated tracking framework. The system is initialized by detections from a pedestrian detector. For each detected person, an independent LS tracker (a *tracklet*) is initialized, which follows this person’s motion in the image space. The LS tracker is kept robust through a series of consistency checks and transmits the tracked person’s bounding box to the high-level tracker after



**Fig. 2.** Level-set segmentation. The contour separates object  $\Omega_f$  from background  $\Omega_b$  in a reference frame given by the warp  $W(\mathbf{x}, \mathbf{p})$ , which is related to the person’s bounding box by the displacement  $V(\mathbf{x}, \mathbf{p}')$ . This contour is the zero level-set of the embedding function  $\Phi$ .

every frame. The high-level tracker in turn converts the bounding boxes to ground plane coordinates and integrates them into physically plausible trajectories using the model selection framework described in Sec. 3.2. During regular operation, the object detector only needs to be activated in regular intervals in order to prevent existing tracklets from degenerating and to start new ones for newly appearing pedestrians. In addition, tracklets can request new detections when they become uncertain. Overall, this results in considerable computational savings, as we will show in Sec. 5.

**Setup.** Similar to previous work on mobile pedestrian tracking [5, 6, 14], we assume a setup of a stereo camera rig mounted on a mobile platform. From this setup, we obtain structure-from-motion (SfM), stereo depth, and a ground plane estimate for every frame. All subsequent processing is then performed only on the left camera stream.

### 3 Algorithmic Components

#### 3.1 Level-Set Tracking

Like [13], we use a probabilistic level-set framework, which first performs a segmentation and in the next frames a rigid registration and shape adaptation. The object shape is defined by the zero level-set of an embedding function  $\Phi(\mathbf{x})$  (Fig. 2) acting on pixel locations  $\mathbf{x}$  with appearance  $\mathbf{y}$ . This level-set is evolved in order to maximize the accordance with learned foreground and background appearance models  $M_f$  and  $M_b$ , while fulfilling certain constraints on the shape of the embedding function and of the contour.

**Segmentation.** The variational level-set formulation for the segmentation consists of three terms which penalize the deviation from the foreground and background model, the deviation of the embedding function from a signed distance function [18], and the length of the contour. A segmentation is achieved by optimizing this energy functional with the following gradient flow [13]:

$$\frac{\partial P(\Phi, \mathbf{p} | \Omega)}{\partial \Phi} = \underbrace{\frac{\delta_\epsilon(\Phi)(P_f - P_b)}{P(\mathbf{x} | \Phi, \mathbf{p}, \mathbf{y})}}_{\text{deviation from fg/bg model}} - \underbrace{\frac{1}{\sigma^2} \left[ \nabla^2 \Phi - \text{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) \right]}_{\text{deviation from signed dist. fct.}} + \underbrace{\lambda \delta_\epsilon(\Phi) \text{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right)}_{\text{length of contour}} \tag{1}$$

where  $P(\mathbf{x}_i|\Phi, \mathbf{p}, \mathbf{y}_i) = H_\epsilon(\Phi(\mathbf{x}_i))P_f + (1 - H_\epsilon(\Phi(\mathbf{x}_i)))P_b$ ,  $\nabla^2$  is the Laplacian,  $H_\epsilon$  is a smoothed Heaviside step function and  $\delta_\epsilon$  its derivative, a smoothed Dirac delta function.  $\Omega = \{\Omega_f, \Omega_b\}$  denotes the foreground/background pixels in the object frame.

$P_f$  and  $P_b$  are the pixel-wise posteriors of the foreground and background models given the pixel appearance. Those models are obtained from the pixels inside and outside the contour during the first segmentation. The segmentation is performed in several iterations and the models are rebuilt in every iteration. In the subsequent tracking steps, the model parameters  $M_f$  and  $M_b$  are only slightly adapted to the current image in order to achieve higher robustness.

**Tracking.** Similar to image alignment, the tracking part aims at warping the next frame such that its content best fits the current level-set. This way, the location of the tracked object is obtained. The warp  $W(\mathbf{x}, \mathbf{p})$  is a transformation of the reference frame with parameters  $\mathbf{p}$ . Any transformation forming a group can be used here, *e.g.* affine transformations. In our application for pedestrian tracking, we currently use only translation+scale. For optimizing the location, the next image is incrementally warped with  $\Delta\mathbf{p}$  until convergence [13]:

$$\Delta\mathbf{p} = \left[ \sum_{i=1}^N \frac{1}{2P(\mathbf{x}_i|\Phi, \mathbf{p}, \mathbf{y}_i)} \left[ \frac{P_f}{H_\epsilon(\Phi(\mathbf{x}_i))} - \frac{P_b}{(1 - H_\epsilon(\Phi(\mathbf{x}_i)))} \right] \mathbf{J}^T \mathbf{J} \right]^{-1} \times \sum_{i=1}^N \frac{(P_f - P_b) \mathbf{J}^T}{P(\mathbf{x}_i|\Phi, \mathbf{p}, \mathbf{y}_i)} \quad (2)$$

with  $\mathbf{J} = \delta_\epsilon(\Phi(\mathbf{x}_i)) \nabla \Phi(\mathbf{x}_i) \frac{\partial W}{\partial \Delta\mathbf{p}}$ , where  $\frac{\partial W}{\partial \Delta\mathbf{p}}$  is the Jacobian of the warp.

**Appearance Models.** [13] only uses color for the foreground and background model. We found that in our application, this yields rather unreliable segmentations for pedestrians, since other people or background structures often contain similar colors. We therefore extend the approach by also including stereo depth information.

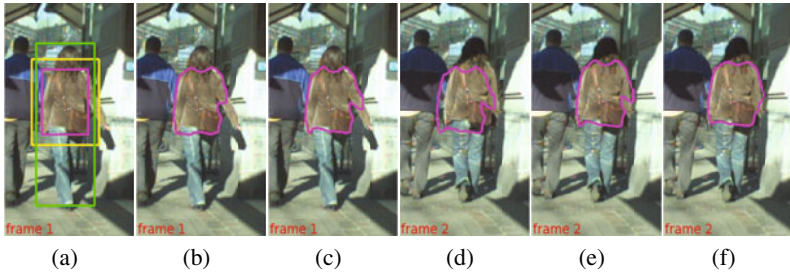
For segmentation, we use the median depth of the foreground area. Unlike the color distribution, the median depth will not stay the same during the following frames. For tracking, we therefore use a simple motion model which computes an expected distance range for each pedestrian according to the last median depth and a maximum velocity. Each depth value in the image is then assigned a probability according to a Gaussian distribution around the median depth or the expected depth, respectively. The color models are represented as L\*a\*b histograms with  $32^3$  bins. The two probabilities for color and depth are individually normalized as in [13] and then merged with a weighting factor  $\alpha$  (set to 0.1 in all of our experiments).

$$P_i = (1 - \alpha)P_{i,color} + \alpha P_{i,depth}, \quad i \in \{f, b\}, \quad (3)$$

### 3.2 Tracking-by-Detection

For the high-level tracker, we use a simplified version of the robust multi-hypothesis tracking framework by [5]. We first describe the basic approach, as it would be applied for pure tracking-by-detection. Section 4 then details how this approach is adapted through the integration with the level-set tracker.

In brief, the approach works as follows. Detected pedestrian locations are converted to 3D world coordinates using the current camera position from SfM together with an



**Fig. 3.** Initialization of the LS tracker: (a) detection box (green), initial object frame (yellow), and initialization of the level-set (magenta); (b,c) evolved level-set after 40 and 150 iterations; (d) level-set transferred to next frame; (e) after warping; (f) after shape adaptation (5 iterations).

estimate of the scene’s ground plane. These measurements are collected in a spacetime volume, where they are integrated into multiple competing trajectory hypotheses. The final hypothesis set is then obtained by applying model selection in every frame.

**Trajectory Estimation.** We model pedestrian trajectories by Kalman filters with a constant-velocity motion model on the ground plane, similar to [6]. When new observations become available in each frame, we first try to extend existing trajectory hypotheses by the new evidence. In addition, we start a new trajectory hypothesis from each new detection and try to grow it by applying a Kalman filter backwards in time through the spacetime volume of past observations. This step allows us to recover lost tracks and bridge occlusions. As a consequence of this procedure, each detection may end up in several competing trajectory hypotheses.

**Model Selection.** For each frame, we try to find the subset of trajectory hypotheses that provides the best explanation for the collected observations. This is done by performing model selection in a Minimum Description Length framework, as in [5]. A trajectory’s score takes into account the likelihood of the assigned detections under its motion and appearance model (represented as a color histogram). Trajectory hypotheses interact through penalties if they compete for the same detections or if their spacetime footprints overlap. For details of the mathematical formulation we refer to [5].

**Assigning Person Identities.** As the model selection procedure may choose a different hypothesis set in each frame, a final step is required in order to assign consistent person IDs to the selected trajectories. This is done by maintaining a list of active tracks and assigning trajectories to them based on the overlap of their supporting observations.

## 4 Combined Tracker

We now present the stages of our combined tracking framework. The difficulty of the street-level mobile tracking task brings with it a number of non-trivial challenges, which we address by consistency checks and carefully modeled interactions between the components of the tracking framework.

**Object Detection.** For pedestrian detection, we apply the widely used HOG detector [19] in the efficient *fastHOG* GPU implementation by [20]. Detections inconsistent with the scene geometry are filtered out by enforcing a ground plane corridor.



**Fig. 4.** Adaptation to lighting changes: (a-c) tracked shape becomes too small due to lighting changes; (d,e) level-set re-initialization is triggered; (f) tracking can continue.

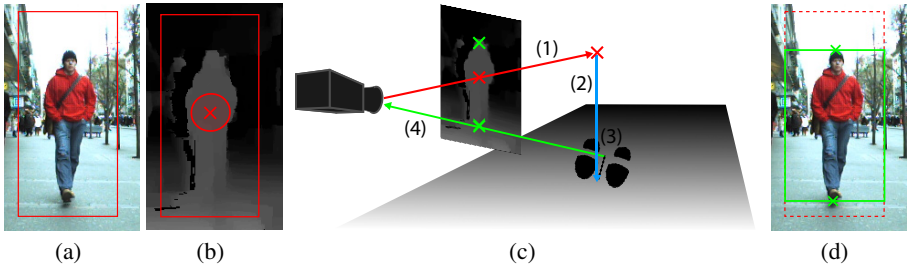
**Level-Set Initialization.** Upon initialization, the LS tracker tries to segment the torso of the person inside a detection box. To this end, a new level-set embedding function is initialized with a rectangular box (see Fig. 3), and the level-set segmentation is iterated for 150 steps. In the next frame, the contour is tracked and the resulting warp is applied to the object frame and the associated detection box in order to obtain the new object position. Afterwards, the level-set shape is adapted for 5 iterations. We track only the person’s torso, since this body part deforms only very little, requiring fewer shape adaptation iterations than tracking the full body. This speeds up level-set tracking and increases the robustness, since it limits the amount of “bleeding” that can occur to similar-colored background pixels. To infer the person’s full extent, we maintain the transformation  $V(\mathbf{x}, \mathbf{p}')$  from the warped reference frame to the original bounding box.

**Multi-Region Handling and Overlap Detection.** When tracking several persons, each of the tracked contours is represented by its own level-set. Even if there are overlaps, the level-sets will not interact directly (as, *e.g.*, in [21]). Instead, we use the stereo depth in order to resolve overlaps. All tracked persons are sorted according to their distance from the camera and the closest person is updated first. All pixels belonging to the resulting segmentation are masked out, such that they cannot be used by the remaining persons.

This leaves us with some persons that are only partially visible, which is in fact the same case as a person leaving the image frame. We developed a method for dealing with partial visibility without losing shape information. As can be seen in eq. (2), only a narrow band of pixels around the contour, which is determined by  $\delta_\epsilon(\Phi)$ , is taken into account for tracking. If pixels are masked out or are outside the image frame, we set  $\delta_\epsilon$  to zero for those pixels, which will result in tracking only the visible part of the contour. Thus, if an object becomes completely visible again, the shape will still fit. Objects are discarded if only a small part of the area inside the contour (50% for person-person occlusions, 20% for occlusions by image borders) remains visible.

**Level-Set Re-initialization.** Lighting changes or similar colors near the object can cause the contour to shrink during tracking (see Fig. 4) or to bleed out during shape adaptation. By periodically updating a tracklet bounding box with new detector bounding boxes, it is possible to identify degenerating shapes based on their size in relation to the bounding box. This is done by first performing the level-set tracking step for adapting the contour to the new image and then matching the tracked location to new detector boxes. If the box overlap (measured by intersection-over-union) is above 0.5,





**Fig. 5.** Depth-based bounding box correction: (a) original bounding box; (b) depth map; (c) correction procedure; (d) corrected bounding box (see text for details)

the detection box is used to update the relationship  $V(\mathbf{x}, \mathbf{p}')$  between box and warp. The level-set contour itself is only updated if its area gets too small or too large with respect to the updated box, or if 20% of its content lie outside the box. Thus, the tracklet integrity is maintained and an ID change is avoided (*c.f.* Fig. 4).

**Consistency Checks.** For robust operation, it is necessary to check the consistency of the tracking results. An object could be occluded, leave the image frame or be lost for other reasons. This may not even have any effect on the convergence of the LS tracker, which might get stuck on some local image structure, resulting in a wrong track. We therefore perform the following checks in order to identify corrupted tracklets. (1) If the object is occluded and only background colors remain, the shape will typically shrink massively within a few frames. If such a case is detected, the tracklet is terminated. (2) We keep track of the median depth inside the tracked contour and react if this value changes too fast. We distinguish two cases here: If the median depth decreases too fast, this indicates an occlusion by another object; if the depth increases too fast, the object was probably lost. We terminate the tracklets in both cases. (3) Finally, objects whose median depth does not fit their ground-plane distance are also discarded. Typically, a failed consistency check indicates a tracking failure and will result in a request for the detector to be activated in the next frame. An exception are cases where an occlusion is “explained” by the high-level tracker (see below), or when the object is close to the image boundary and is about to leave the image.

**Depth-based Bounding Box Correction.** Level-set (re-)initialization and high-level 3D trajectory integration require accurately aligned bounding boxes. In general, the HOG detector however yields detections with a certain border area. Similarly, the boxes provided by the LS tracker may drift due to articulations and shape changes of the level-set contour and need to be corrected. We therefore apply the following correction procedure both to new detections and after each level-set tracking step. Starting from the original bounding box (Fig. 5(a)), we first compute the median depth around the bounding box center (Fig. 5(b)). We then determine the corresponding 3D point using the camera calibration from SfM and project it onto the ground plane (Fig. 5(c), steps(1)+(2)). We add a fixed offset in the viewing direction in order to determine the person’s central foot point, and finally project the resulting 3D point back to the image (Fig. 5(c), steps (3)+(4)). This determines the bottom line of the corrected bounding box. The top line is found by searching for the highest point inside the bounding box

that is within 0.5m of the median depth (Fig. 5(d)). As a final step, we verify that the resulting bounding box aspect ratio is in the range  $[\frac{1}{3}, \frac{2}{3}]$ . Bounding boxes falling outside this range are rejected as likely false positives.

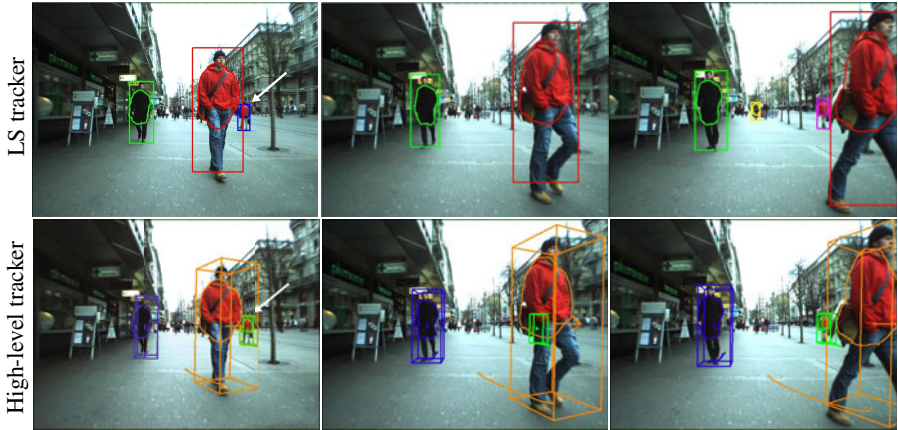
**Requesting New Detections.** New detections are requested in the following cases: (1) if a tracklet has not received an updated detection in the last  $k$  frames; (2) if a tracking failure cannot be explained by an occlusion or by the tracked person leaving the image; (3) if no request has been issued for  $k$  frames (e.g., since no object is currently tracked). A tracklet will not request new detections if it is close to the image boundary, as the chance for finding a detection there would be small. If a tracklet receives no updated detection despite its request, it will repeat the request, but will continue to be tracked as long as it passes the consistency and depth correction checks.

**Integration with High-Level Tracker.** The high-level tracker’s task is to integrate the tracklet bounding boxes into physically plausible 3D trajectories. This is done by first creating an *observation* at each tracked person’s 3D foot point and then associating this observation to trajectory hypotheses. The overall procedure is similar to the general tracking-by-detection framework described in Sec. 3. However, we make the following changes in order to account for the additional information provided by the LS tracker.

Since we already know the tracklet identity of each observation from the LS tracker, we can use this information in order to simplify data association. Thus, we first try to extend each existing trajectory hypothesis by searching for an observation matching the trajectory’s currently followed tracklet ID in a gating area around the Kalman filter prediction. If such an observation can be found, it will directly be associated with the trajectory. Note that in this case, only the motion model is considered; the appearance is assumed to be correct due to the association performed by the LS tracker. In case no observation with the correct tracklet ID can be found, we try to find the best-matching observation under the trajectory’s motion and appearance model (again within a gating area determined by the Kalman filter uncertainty). If such a new observation can be found, the trajectory takes on the new tracklet ID, thus connecting the two tracklets. This latter case can occur if the LS tracker diverges and fails the consistency checks (in which case the tracklet will be terminated), if the tracked bounding box is rejected by the depth correction (in which case the tracklet may persist for up to  $k$  frames and can be recovered), or if the tracked object is occluded or leaves the image.

In addition to the above, each observation is used to start a new trajectory hypothesis, which searches backwards in time in order to find a potentially better explanation for the observed data. This makes it possible to automatically create tracks for newly appearing persons and to correct earlier tracking errors. The final set of accepted tracks is then obtained by performing model selection, as described in Section 3.2.

**Tracking through Occlusions.** As motivated above, a main advantage of the image-based low-level tracker, compared to a pure tracking-by-detection approach, is that it simplifies data association, thus making it easier to integrate observed pedestrian locations into valid tracks. The image-based tracklet generation will however fail when the tracked person gets occluded, which often occurs in practice. This is a limitation of any image-based tracking approach. While strategies can be devised to cope with short-term occlusions at the image-level, they would make this component unnecessarily complex.

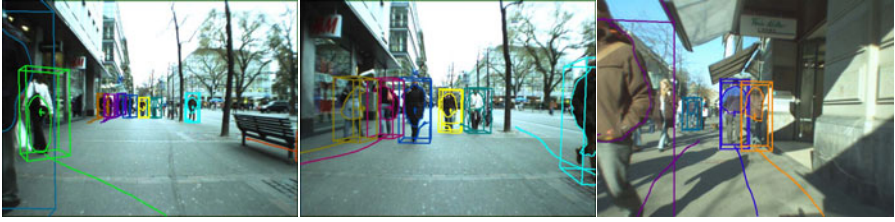


**Fig. 6.** Example for the occlusion handling process: (top row) contours tracked by the LS tracker; (bottom row) output of the high-level tracker. When the distant person is temporarily occluded, its LS tracklet is terminated. As soon as the occlusion is over, a new tracklet is started. The high-level tracker connects both tracklets through the occlusion and maintains the person’s identity.

In our approach, we instead address this issue by explicit occlusion handling on the high-level tracker’s side. In order to bridge short-time occlusions, we keep potentially occluded trajectories alive for up to 15 frames and extrapolate their position on the ground plane using the Kalman filter. Since the latter’s positional uncertainty grows with the absence of observations, the corresponding person can likely be associated to the predicted trajectory again when reappearing from the occlusion.

In addition, the high-level tracker can predict person-person occlusions and reinitialize the image-based tracker when those are over. For this, we backproject the predicted 3D bounding box of each tracked person into the image and compute the bounding box overlap using the intersection-over-union criterion. If the overlap is larger than 0.5, then an occlusion is likely to occur. This information is stored together with the occluded trajectory and is transmitted to the corresponding LS tracklet, which will typically be terminated 1-2 frames later when the consistency check fails. When the corresponding object is predicted to become visible again a few frames later, the object detector is fired in order to recover the person with as little delay as possible. This “safe termination” and subsequent new tracklet generation strategy proved to be robust in our experiments. It is similar in spirit to the *track-suspend-fail* strategy proposed in [15], but our approach extends the idea through the integration of the robust multi-hypothesis tracking framework.

Fig. 6 shows an example where this occlusion handling process is used in practice. Cued by the occlusion prediction and the failed depth consistency check, the LS tracklet is terminated in order to avoid degeneracies (which would be likely in this case due to the similar color distributions). On the high-level tracker’s side, the trajectory is however extrapolated through the occlusion. As soon as the occluded person becomes visible again, the object detector is fired again in order to initialize a new LS tracklet, which is correctly associated to the trajectory, maintaining the person’s identity.



**Fig. 7.** Examples demonstrating our approach’s capability to continue tracking persons close to the camera and/or the image borders, where object detection is no longer applicable

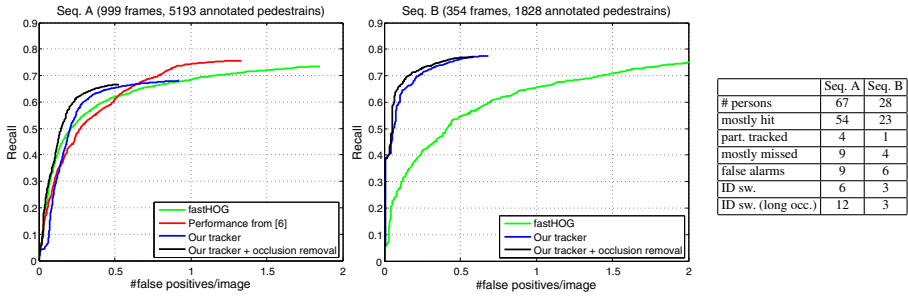
## 5 Experimental Results

**Datasets.** We evaluated our approach on two challenging sequences from the Zurich Mobile Pedestrian corpus generously provided by the authors of [6]. We used the sequences BAHNHOF (in the following: “Seq. A”) and SUNNY DAY (“Seq. B”). Both sequences were captured with a stereo rig (13-14fps, 640x480). Seq. A (999 frames, with 5193 annotated pedestrians of  $\geq 60$  pixels height) was taken on a crowded sidewalk on a clouded day. Seq. B (999 frames, 354 of which are annotated with 1867 annotations) was captured on a sunny day and contains strong illumination changes. Both sequences come with stereo depth maps, structure-from-motion localization, and ground plane estimates. Similar to [6], we upscale all images to twice their original resolution in order to detect also pedestrians at larger distances. Using the upscaled images, fastHOG performed at 2-3fps (10fps for original images). In contrast to [5, 6], we however only use the left camera stream for detection and tracking, thus reducing the necessary processing effort. All system parameters were kept the same throughout both sequences.

**Tracking Performance.** Figure 7 shows qualitative results of our approach, demonstrating its capability to continue tracking persons that appear close to the camera or that are partially occluded by the image boundaries. This is a fundamental advantage our tracking framework can offer over pure tracking-by-detection approaches.

In order to assess our approach’s performance quantitatively, we adopt the evaluation criteria from [6] and measure the intersection-over-union of tracked person bounding boxes and annotations in every frame. We accept detections having an overlap greater than 0.5 as correct and report the results in terms of *recall vs. false positives per image* (fppi). Fig. 8 shows the resulting performance curves when we set the maximum re-initialization interval to  $k = 5$  frames (in blue), together with the baseline of fastHOG (in green). As can be seen, our approach achieves good performance, reaching 65% and 76% recall at 0.5 fppi for Seq. A and Seq. B, respectively. As the bounding box criterion penalizes the tracker’s property of predicting a person’s location through occlusions (since those cases are not annotated in the test data), we additionally provide the performance curve when filtering out tracked bounding boxes which are more than 50% occluded by other boxes (in black). This results in an additional improvement.

For comparison, we also provide the performance curve reported by [6] on Seq. A, which is also based on HOG detections (shown in red, no such curve is available for Seq. B). This approach integrates detections from both camera streams and thus obtains



**Fig. 8.** (left) Quantitative tracking performance of our approach compared to different baselines. (right) Track-level evaluation according to the criteria by [7].

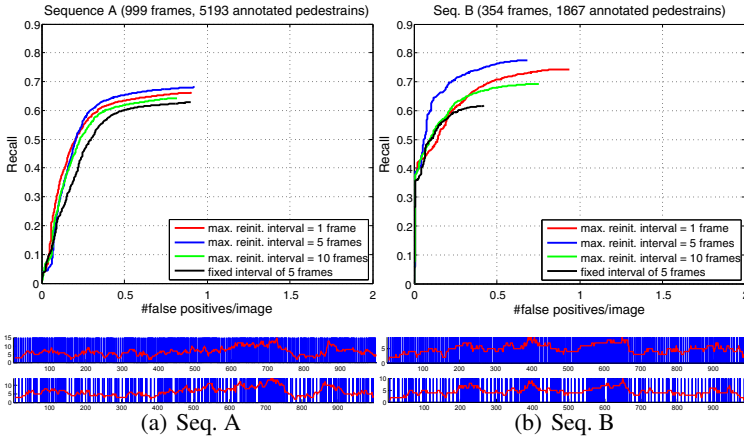


**Fig. 9.** Example tracking results of our approach on several challenging test sequences

a higher recall. Its performance should be compared to our blue curve, since no occlusion removal was performed in [6]. Still, it can be seen that our approach achieves better performance in the high-precision range, despite only using a single camera stream. This is a result of the better data association provided by the image-level tracklets.

Fig. 8 (right) also reports a track-level evaluation according to the criteria by [7], showing that most pedestrians are correctly tracked and only few ID switches occur. Fig. 9 shows results of our combined tracker for both test sequences and visualizes the obtained level-set contours. The corresponding result videos are provided on [www.mmp.rwth-aachen.de/projects/eccv2010](http://www.mmp.rwth-aachen.de/projects/eccv2010). Our system is able to track most of the visible pedestrians correctly in a very busy environment with many occlusions.

**Efficiency Considerations.** One of our goals was to reduce the dependence on the costly detector. Even though efficient GPU implementations are now available for HOG (e.g. [20]), the framerate is still not sufficient for real-time operation in a pure tracking-by-detection context. In addition, the excessive power consumption of GPUs is a major restriction for their use in mobile robotics applications. In contrast, the level-set tracking approach employed here can be very efficiently implemented on regular CPUs. [13]



**Fig. 10.** (top) Tracking performance for the two test sequences when varying the maximum re-initialization interval; (bottom) Frequency of detector activations for both sequences for an interval of 5 (first) and 10 (second) frames. The red curve shows the number of tracked pedestrians.

report a framerate of 85Hz for tracking a single target of size  $180 \times 180$  pixels in their implementation. In our application, we track targets at a lower resolution of  $80 \times 100$  pixels and therefore expect even faster performance once our code is fully optimized.

An important consideration in this respect is how often the pedestrian detector needs to be activated for robust tracking performance. Our approach lets the LS tracker request detections whenever required, but enforces a maximum re-initialization interval of  $k$  frames. Fig. 10 shows the effective frequency of detector activations when setting this interval to  $k \in \{1, 5, 10\}$ , together with the resulting tracking performance. A setting of  $k = 5$  provides the best tracking quality with a detector activation on average every 1.66 frames. By increasing the maximum interval to 10 frames, the detector activation rate falls to every 2.71 frames at a small loss in recall that is still comparable to [6] at 0.5 fppi. Considering that [6] performed detection in both camera streams, our approach thus requires 5.42 times less detector activations. Finally, we show the performance when activating the detector at a fixed interval of 5 frames, without additional requests. This results in a small drop in recall, but still yields good overall performance.

## 6 Conclusion

We have presented an integrated framework for mobile street-level multi-person tracking. Our approach combines the advantages of a fast segmentation-based tracker for following individual persons with the robustness of a high-level multi-hypothesis tracking framework for performing longer-term data association. As our experiments have shown, the approach reaches state-of-the-art performance, while requiring fewer detector evaluations than conventional tracking-by-detection approaches. Our results open several interesting research perspectives. The requested detector activations for tracklet re-initialization could be restricted to the tracklet’s immediate neighborhood, thus resulting in further speedups. In addition, the obtained level-set segmentation could be a possible starting point for articulated tracking that we plan to explore in future work.

**Acknowledgments.** This project has been funded, in parts, by the EU project EUROPA (ICT-2008-231888) and the cluster of excellence UMIC (DFG EXC 89). We thank C. Bibby and I. Reid for valuable comments for the level-set tracking and for making their evaluation data available.

## References

1. Stauffer, C., Grimson, W.: Adaptive Background Mixture Models for Realtime Tracking. In: CVPR'99 (1999)
2. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian Detection: A Benchmark. In: CVPR'09 (2009)
3. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
4. Andriluka, M., Roth, S., Schiele, B.: People Tracking-by-Detection and People Detection-by-Tracking. In: CVPR'08 (2008)
5. Leibe, B., Schindler, K., Van Gool, L.: Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. PAMI 30, 1683–1698 (2008)
6. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Robust Multi-Person Tracking from a Mobile Platform. PAMI 31, 1831–1846 (2009)
7. Wu, B., Nevatia, R.: Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Part Detectors. IJCV 75, 247–266 (2007)
8. Huang, C., Wu, B., Nevatia, R.: Robust Object Tracking by Hierarchical Association of Detection Responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
9. Zhang, L., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: ECCV'08 (2008)
10. Zhao, T., Nevatia, R., Wu, B.: Segmentation and Tracking of Multiple Humans in Crowded Environments. PAMI 30, 1198–1211 (2008)
11. Grabner, H., Grabner, M., Bischof, H.: Real-Time Tracking via On-line Boosting. In: BMVC'06 (2006)
12. Cremers, D., Rousson, M., Deriche, R.: A Review of Statistical Approaches to Level Set Segmentation Integrating Color, Texture, Motion and Shape. IJCV 72, 195–215 (2007)
13. Bibby, C., Reid, I.: Robust Real-Time Visual Tracking using Pixel-Wise Posteriors. In: ECCV'08 (2008)
14. Gavrilu, D., Munder, S.: Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle. IJCV 73, 41–59 (2007)
15. Kaucic, R., Perera, A., Brooksby, G., Kaufhold, J., Hoogs, A.: A Unified Framework for Tracking through Occlusions and Across Sensor Gaps. In: CVPR'05 (2005)
16. Tao, H., Sawhney, H., Kumar, R.: Object Tracking with Bayesian Estimation of Dynamic Layer Representations. PAMI 24, 75–89 (2002)
17. Ge, W., Collins, R.: Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In: BMVC'08 (2008)
18. Li, C., Xu, C., Gui, C., Fox, M.: Level Set Evolution without Re-initialization: A New Variational Formulation. In: CVPR'05 (2005)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR'05 (2005)
20. Prisacariu, V., Reid, I.: fastHOG – a Real-Time GPU Implementation of HOG. Technical Report 2310/09, Dept. of Engineering Science, University of Oxford (2009)
21. Brox, T., Weickert, J.: Level Set Based Image Segmentation with Multiple Regions. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 415–423. Springer, Heidelberg (2004)

# Closed-Loop Adaptation for Robust Tracking

Jialue Fan, Xiaohui Shen, and Ying Wu

Northwestern University  
2145 Sheridan Road, Evanston, IL 60208  
{jfa699,xsh835,yingwu}@eecs.northwestern.edu

**Abstract.** Model updating is a critical problem in tracking. Inaccurate extraction of the foreground and background information in model adaptation would cause the model to drift and degrade the tracking performance. The most direct but yet difficult solution to the drift problem is to obtain accurate boundaries of the target. We approach such a solution by proposing a novel closed-loop model adaptation framework based on the combination of matting and tracking. In our framework, the scribbles for matting are all automatically generated, which makes matting applicable in a tracking system. Meanwhile, accurate boundaries of the target can be obtained from matting results even when the target has large deformation. An effective model is further constructed and successfully updated based on such accurate boundaries. Extensive experiments show that our closed-loop adaptation scheme largely avoids model drift and significantly outperforms other discriminative tracking models as well as video matting approaches.

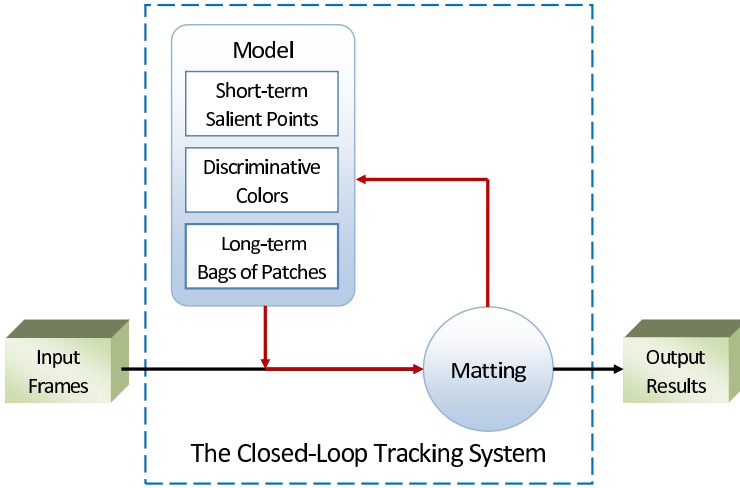
## 1 Introduction

Object tracking is a fundamental task in computer vision. Although numerous approaches have been proposed, robust tracking remains challenging due to the complexity in the object motion and the surrounding environment. To reliably track a target in a cluttered background, an adaptive appearance model that can discriminate the target from other objects is crucial. It has been shown that in many scenarios context information can be adopted to increase the discriminative power of the model [1,2].

One way of incorporating context information is to find auxiliary objects around the target and to leverage the power of these objects to collaboratively track the target [19]. However, these methods require the presence of objects whose motion is consistently correlated to the target, which may not be satisfied sometimes. Another way is to extract the features of the background around the target, are then use them to enhance the distinction of the target against the background, either by feature selection [1], or by training classifiers [21].

One critical issue that is rarely discussed in these methods is the degradation of the model caused by the inaccuracy in the estimation of the foreground and background. Most commonly the foreground and background are divided by a bounding box or a region around the location of the target. No matter how tight





**Fig. 1.** The framework of closed-loop adaptation for tracking

the region is, such a partition is too rough because some background regions are treated as part of the foreground, especially when the location of the target is not precise or the target is occluded. Accordingly, the updated model would gradually be degraded and thus cause drift. Grabner *et al.* [18] proposed an online semi-supervised boosting method to alleviate drift, and Babenko *et al.* [3] introduced multiple instance learning to handle the problem. Despite such efforts, an accurate boundary that clearly divides the target from the background is still desirable.

To obtain a clear boundary of the foreground, one effective way is to perform matting based on some prior information, which has been shown very successful in estimating the opacity of the foreground. The boundary can then be easily extracted from the opacity map. However, matting has never been combined with tracking before because of the gap that matting needs user interaction while tracking requires automatic processing. Video matting, although using some tracking techniques (*e.g.* optical flow) to lighten the burden of human efforts, still needs a large amount of user input and can not meet specific demands of object tracking such as automatic processing, low resolution and occlusion handling. In this paper, we bridge this gap by automatically providing suitable scribbles for matting during the tracking process and make matting work very well in the tracking scenario. Furthermore, we propose a practical model adaptation scheme based on the accurate object boundary estimated by matting, which largely avoids the drift problem. Such an interplay of matting and tracking therefore forms a **closed-loop** adaptation in an object tracking system, as shown in Fig. 1. Compared to other tracking approaches, our closed-loop tracking system has the following contributions and advantages:

1. We address the automatic scribble generation problem for matting in the tracking process. A coarse but correct partition of foreground and background is estimated during tracking, which is then used to automatically generate suitable scribbles for matting. The supply of scribbles is non-trivial. A small false scribble may lead to matting failure, while deficient scribbles could also impair the performance. In our system, the generation of scribbles is designed carefully to be correct and sufficient, which can yield comparable matting results to the methods based on user input.
2. We construct a simple but practical model for tracking, which not only captures short-term dynamics and appearances of the target, but also keeps long-term appearance variations, which allows us to accurately track the target in a long range under various situations such as large deformation, out of plane rotation and illumination change, even when the target reappears after complete occlusion.
3. Unlike other methods that tried to alleviate the aftereffects caused by inaccurate labeling of the foreground, we successfully extract the accurate boundary of the target and obtain refined tracking results based on alpha mattes. Under the guidance of such a boundary, the short-term features of the model are updated. Moreover, occlusion is inferred to determine the adaptation of the long-term model. Benefiting from the matting results, our model adaptation largely excludes the ambiguity of foreground and background, thus significantly alleviating the drift problem in tracking. Besides, object scaling and rotation can also be handled by obtaining the boundary.

## 2 Related Work

Object tracking has been an active research area since early 1980s and a large number of methods were proposed during the last three decades. In the perspective of model design and update in tracking, early works tended to construct the model by describing the target itself [22,23], while recently the adoption of context information has become very popular [1,2,19,21,20,4,5,14].

The modeling of spatial context in these methods can be categorized to two levels: higher object level and lower feature level. At the higher level, the interactions between different targets are explored in multiple target tracking, either by a Markov network [5] or by modeling their social behaviors [4]. Such interactions are further extended to the auxiliary objects around the target [19]. By finding the auxiliary objects whose motion is consistently correlated to the target at a certain short period, it can successfully track the target even if the appearance of the target is difficult to discriminate. However, such auxiliary objects are not always present, which makes those methods sometimes not applicable.

At the lower level, the features of the background around the target are utilized without analyzing their semantic meanings. Feature selection can be performed by choosing the most discriminative ones between the target and its background, which is first proposed in [1]. Avidan [21] trained an ensemble of classifiers by treating the target as positive samples and the background as negative ones. These methods, however, more or less suffer from the inaccuracy in the

estimation of the foreground and background, which obscures the discrimination of their model and eventually leads to drift.

The drift problem was discussed in [6], in which they proposed a partial solution for template update. Grabner *et al.* [18] proposed an online semi-supervised boosting method, and Babenko *et al.* [3] introduced multiple instance learning to avoid the drift of positive samples. However, these methods all focused on the alleviation of drift caused by foreground/background labeling errors. If an accurate boundary of the target can be obtained, such errors would be mostly reduced.

Image segmentation is one way to extract and track the object boundaries. Ren *et al.* [16] combined spatial and temporal cues in a Conditional Random Field to segment the figure from the background in each frame, and Yin *et al.* [15] modified the CRF by adding shape constraints of the target. However some tracking scenes may have cluttered backgrounds, which cause difficulties to directly extract accurate boundaries using segmentation techniques.

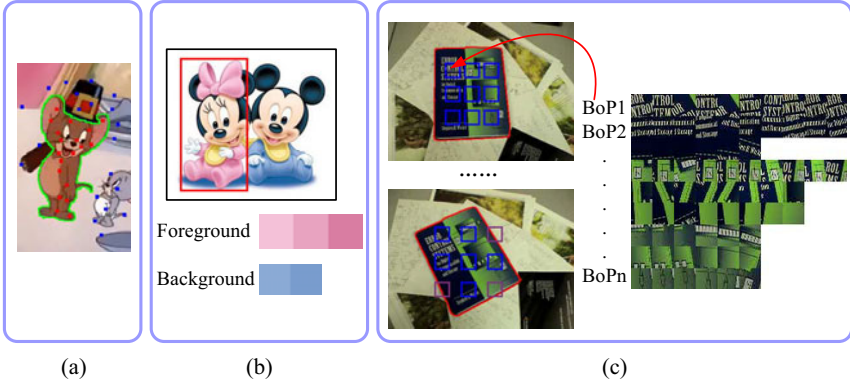
Compared with image segmentation, alpha matting tries to exploit the linear compositing equations in the alpha channel instead of directly handling the complexity in natural images, therefore may achieve better foreground/background separation performance based on a moderate amount of user input [7]. [8] provided an extensive review of recent matting techniques. Matting is further extended to videos by combining motion estimation approaches such as optical flow and background estimation [9,10]. But video matting can not be directly used in tracking, as they always need user interaction, which is not suitable in automated tracking methods. Moreover, they can not well handle objects with fast deformations and occlusions, which are very common in most tracking scenarios. To the best of our knowledge, our method is a first attempt to combine matting with tracking to provide shape boundary of the target and to handle occlusion.

### 3 Model Description

By incorporating the properties of discriminative models and descriptive models, our tracking model tries to discriminate the foreground from the background as well as maintain a long-term description for the target's appearance. Apart from the basic dynamics, the model is composed of three main components.

**Short-term salient points.**  $S_f$  denotes a set of salient points that are extracted from the foreground, while  $S_b$  is a set of salient points detected from the surrounding background near the target, as shown in Fig. 2(a). Currently SIFT [11] are used as salient points. Salient points are tracked in a short time period and used to generate scribbles for matting and estimate the dynamics of the model.

**Discriminative colors.** Color is another useful clue to discriminate the foreground from the background. We select the most discriminative colors for the foreground and the background respectively. Given a frame with known foreground and background (either by manual initialization at the first frame or



**Fig. 2.** Our model for tracking. (a) Short-term salient points, (b) discriminative color lists, (c) long-term bags of patches.

by refined tracking results at the following frames), we can obtain the discriminative color list of the foreground  $C_f$  based on the log-likelihood ratio of the foreground/background color histogram<sup>1</sup>. We can get a similar list of the background  $C_b$ , and maintain these two color lists respectively. Figure 2(b) gives us an example. In Fig. 2(b), the pink color is the most distinctive one for the foreground, while light blue is distinctive for the background. White, black and yellow exist in both the foreground and background. Therefore neither of  $C_f$  and  $C_b$  chooses them as discriminative colors. Such a description, although simple, is observed very powerful to detect the foreground and the background.

**Long-term bags of patches.** We also constructed a long-term model to preserve the appearance variation of the target in a long range, which helps locate the target under occlusion and deformation. Given a target region, we divide it to a  $M \times N$  grid. For example, in Fig. 2(c), the grid is  $3 \times 3$ . At each crosspoint of the grid, a **patch** is cropped and recorded<sup>2</sup>. Therefore we have many patches with different time stamps at each crosspoint, which captures the variation in the local appearance at a relatively fixed position of the foreground. We call the set of all the patches at the same crosspoint a **bag** of patches  $BoP_i$ . For example, in Fig. 2(c),  $BoP_1$  captures the long-term appearance at the top-left corner of the target. The adaptations of those bags are performed independently by foreground matching and occlusion inference, which avoids false update due to partial occlusion. The geometric information (normalized by target size) of these bags of patches is also implicitly encoded by their relative positions.

At each frame, the short-term features in the model are used to generate foreground/background scribbles for matting, and the long-term model is utilized to locate the object when it is under severe occlusion and the short-term features

<sup>1</sup> We divide the color to 1024 bins in HSV space(16 bins, 8 bins and 8 bins in the H, S and V channels respectively), and then get the color histogram.

<sup>2</sup> The patch size is  $K \times K$ . In practice we choose  $K = 25$ .

are not reliable. Once the accurate foreground boundary is determined by matting, all the components of the model will be updated accordingly, which will be introduced in the next two sections.

## 4 The Closed Loop: From Coarse Tracking to Matting

Given an input frame, we first use our model to perform coarse tracking. *i.e.*, locate the target and obtain a coarse but correct partition of the foreground and background. Based on such a partition, scribbles are automatically generated for matting. The matting results heavily rely on the prior information of the foreground and background. A false labeling of foreground or background may cause a drastic erroneous matte. We carefully design the scribble generation scheme to avoid false labeling and to yield good alpha mattes.

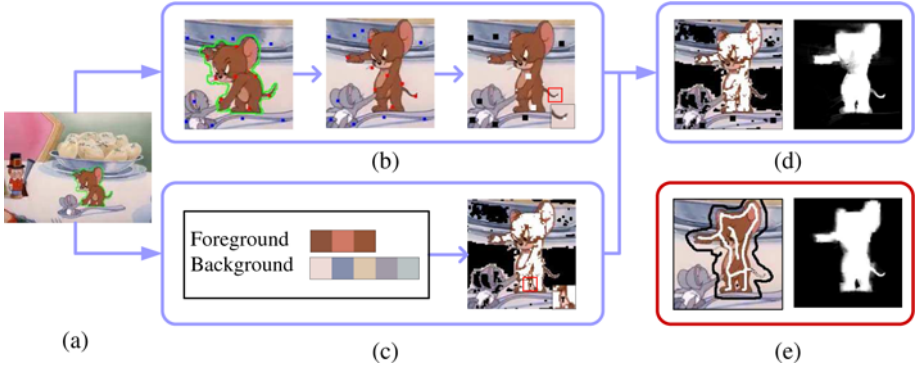
### 4.1 Short-Term Coarse Tracking

From frame to frame, we use two types of features to detect the foreground and background and then generate scribbles: salient points and homogenous regions.

**Salient points.** Consider that  $S_f$  and  $S_b$  are the sets of salient points extracted from the foreground and its neighboring background at the previous frame  $f^{t-1}$  respectively, and  $S'_f$  and  $S'_b$  are the corresponding salient point sets at the current frame  $f^t$ . First we perform SIFT matching between  $S_f$  and  $S'_f$ . However, we can not guarantee that the salient points at  $f^{t-1}$  are still salient at  $f^t$ . Therefore, for those points in  $S_f$  that do not find their matching points in  $S'_f$ , they are tracked by calculating SSD and gradient-based search to find new locations at  $f^t$  [13]. At last all the points in  $S_f$  will have matched locations at the current frame. Small image regions that cover these new locations are then labeled as the foreground. Similarly, we track all the salient points in  $S_b$  and label the regions covering their new locations as the background, as we can see in Fig. 3(b). The sizes of scribbles depend on the positions of salient points and their matching scores. If a salient point is far from the object boundary at  $f^{t-1}$  and its matching score is relatively high, the corresponding scribble will be relatively large, otherwise it will be small to avoid false labeling.

It is worth notice that in our coarse tracking process, the tracking results of these salient points are not necessary to be very accurate. *It only requires that  $S_f$  still stay in the object and  $S_b$  remain in the background*, which is robust to some fluctuations in salient points tracking. In our experiments we found that such requirements are easily satisfied by the tracking results.

**Discriminative color regions.** Although the regions with salient points are labeled, there are still large uncertain regions around the object, some of which are very discriminative in color space. Therefore we choose discriminative color regions as additional scribbles. Consider that the discriminative color lists  $C_f$  and  $C_b$  have been updated at  $f^{t-1}$ . At  $f^t$ , we use these two lists to detect foreground discriminative color regions and background regions at the possible locations of



**Fig. 3.** Short-term coarse tracking. White regions denote foreground scribbles, while black ones denote background scribbles. (a) The boundary of the target at previous frame, (b) generating scribbles by salient point tracking, (c) Generating scribbles by the discriminative color lists, (d) final scribbles and estimated alpha matte, (e) matting result by user input.

the target. For each pixel, if its color is the same as one color in foreground discriminative color list  $C_f$ , it will be labeled as foreground. Similarly, the pixels with the same colors as in  $C_b$  are marked as background, as shown in Fig. 3(c).

The scribbles provided by salient points and the ones provided by discriminative color regions are good complements to each other. As we can see in the red square region in Fig. 3(b) (an enlarged view is provided in the lower-right corner), salient points can be detected on the tail of Jerry. And in Fig. 3(c), the region between two legs of Jerry is marked as background, where no salient points exists. Combining two categories of scribbles, the final scribbles for matting are drawn in Fig. 3(d), which ensures to produce a satisfying matte.

Given such scribbles, standard matting methods can be adopted to estimate the matte of current frame. Here we use the closed-form solution proposed in [7]. As we can see in Fig. 3(d), Our scribbles are already good and sufficient to estimate a good matte, which is very competitive against the matting result based on user input in Fig. 3(e).

## 4.2 Long-Term Target Locating

In most situations, the tracking results of salient points and the identification of discriminative colors are satisfying to help generate a good alpha matte. However, in some cases, such a method is not applicable, especially when the target is severely occluded and no sufficient salient points can be provided, or when the target reappears from complete occlusion. To address those problems, we use our long-term bag-of-patches model to locate the target.

Our model matching approach is based on an underlying assumption: no matter whether the target is severely occluded or first reappears, only a small part of the target is visible. Therefore, only one or two bags in our model are in the

foreground at this time. That assumption significantly simplifies our matching scheme. We sequentially use one of the bags to search the space. Each maintained patch in that bag is used to find their most matched patch in the searching space (the patch with the best SSD matching score). Among those matching scores, the highest one is recorded as the matching confidence of that bag. We identify the bag with the highest confidence as the searching results. *i.e.*, the matched patch by that bag is labeled as the foreground, and the location of the model is also determined by that patch. For example, in Fig. 8,  $BoP_7$  has the highest matching confidence at Frame 720, the target is then located according to  $BoP_7$ .

If the target is severely occluded, we can still infer its possible location according to previous tracking results, and the target can be searched in that possible region. If the target reappears from complete occlusion, then searching in the whole space may be needed. If the search space is too large, it is quite computationally expensive. Therefore we propose a coarse-to-fine method to relocate the target. We perform search every 5 pixels and find the best one, then using gradient-based SSD matching to find the local optimum. We observed that the performance is sufficiently good in experiments by this fast method.

After locating the target, the matched patch provides a foreground scribble for matting. Discriminative color detection is further performed again to generate additional scribbles around the target. Matting thus can be successfully performed using these scribbles.

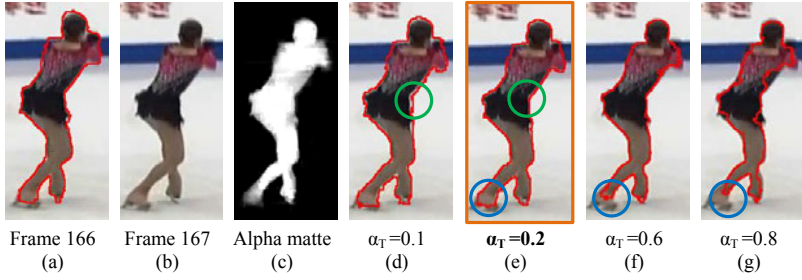
## 5 The Closed Loop: From Matting to Refined Tracking

In the other half loop of our system, estimated alpha mattes are first adopted to refine tracking results (*i.e.* to obtain the accurate boundary and the dynamics of the target). Each component in our model can then be sequentially updated based on the clear foreground and background.

### 5.1 The Boundary

The alpha matte is a continuous map of opacity  $\alpha$ . The  $\alpha$  values near the boundary of the target are hardly 0 or 1 but some values between them. Therefore, to remove such ambiguity and to obtain a clear shape boundary of the target, a certain  $\alpha$  threshold  $\alpha_T$  must be chosen to cut this map.

The shape boundary of the previous frame is used as the guide to determine  $\alpha_T$ . For example, given the boundary at Frame 166 in Fig. 4(a) and the estimated matte at Frame 167 in Fig. 4(c), by setting different thresholds, we can obtain different shape boundaries, as shown in Fig. 4(d)-(g). We assume that although the target may have large deformation, its shape in two consecutive frames should not be too different. Therefore the one having the maximum likelihood with the previous shape boundary is chosen as the boundary at the current frame. We used the contour matching method in [12] to calculate the likelihood because of its computational efficiency. The final chosen  $\alpha_T$  is 0.2, and the boundary of the target determined by alpha matting is shown in Fig. 4(e). Compared with



**Fig. 4.** Determining the boundary from estimated matte. Under the guidance of the boundary at Frame 166 in (a), the threshold is selected to be 0.2, and the boundary at Frame 167 is given in (e).

this selected threshold, a smaller  $\alpha_T$  takes some background as foreground (the region circled using green color in Fig. 4), while a larger  $\alpha_T$  tends to exclude some true foreground, as shown in the blue circle region in Fig. 4.

### 5.2 Estimating the Dynamics of the Target

The dynamics of the model are estimated from the motions of the salient points. According to the positions of the salient points at the current frame and their corresponding positions at the previous frame, their motion vectors  $\mathbf{v}_i$  between these two frames are easily calculated, as shown in Fig. 5(a). Based on  $\mathbf{v}_i$ , a dominant motion of the entire target can be estimated. We use Parzen window method to generate a 2-D density map of salient point motion vectors.

$$f(\mathbf{v}) = \frac{1}{nh^2} \sum_{i=1}^n K\left(\frac{\mathbf{v} - \mathbf{v}_i}{h}\right) \tag{1}$$

where  $h$  is the bandwidth, and  $K(x)$  is the window function. Here we set  $h = 3$  and  $K(x)$  is a standard Gaussian function with mean zero and covariance matrix  $\sigma I$  ( $I$  is an identity matrix). If the motion vectors of salient points present coherence, which means the entire target is also moving with a dominant motion, the motion density map must have a sharp peak (Fig. 5(b)). Let  $\mathbf{v}_m$  denote the motion with the maximum density. We calculate the motion density around  $\mathbf{v}_m$ :

$$P(\mathbf{v}_m) = \int_{\|\mathbf{v} - \mathbf{v}_m\| < 1} f(\mathbf{v}) d\mathbf{v} \tag{2}$$

If  $P(\mathbf{v}_m) \geq \beta$ , the peak is considered very sharp, and  $\mathbf{v}_m$  is the dominant motion of the entire target. The location of the target is then determined by:

$$L^t = L^{t-1} + \mathbf{v}_m \Delta t \tag{3}$$

where  $L^{t-1}$  is the location of the target at previous frame, and  $\Delta t = 1$ .



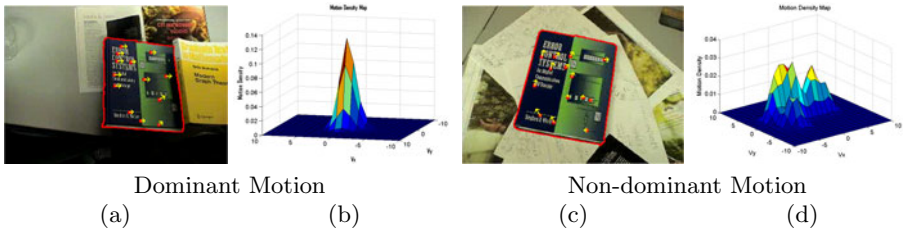


Fig. 5. Dominant motion estimation

If  $P(\mathbf{v}_m) < \beta$ , it indicates that the motions of salient points are not coherent, and the target may be rotating or deforming without a dominant motion, as in Fig. 5(c). Therefore, the motion estimated by salient points is not reliable. In that case, we directly use the long-term model to match the current foreground and find the location of the model, as introduced in Sect. 4.2.

### 5.3 Model Updating

After obtaining the clear foreground, all the components in our model are updated.

**Updating salient points.** Salient points are short-term features, therefore we directly re-sample new points in the foreground and the neighboring background to get obtain  $S_f$  and  $S_b$ .

**Updating discriminative colors.** During tracking, the background color may largely change, while the foreground may also vary due to deformation and self occlusion. Therefore, the discriminative color lists should be updated to remove invalid colors and add new discriminative colors.

Once the target is located and the boundary is estimated, we first get the color histograms of the foreground and background. Discriminative colors for the foreground and the background are then extracted respectively by calculating the log-likelihood ratio of these two histograms, as introduced in Sect. 3. For each extracted foreground discriminative color at current frame, we compare it with  $C_f$  and  $C_b$ . There are three cases:

1.  $C_b$  contains the same color, *i.e.*, one of the color features in  $C_b$  and this newly extracted discriminative color fall into the same color bin. It means that this color feature is no more discriminative for the background, and thus will be removed from  $C_b$ .
2.  $C_b$  does not contain this color while  $C_f$  does, then this color is discriminative for the foreground but already exists in  $C_f$ . No update will be performed.
3. Neither of  $C_b$  and  $C_f$  has the same color. Apparently this color feature is a new discriminative color for the foreground and will be added to  $C_f$ .

Similarly, we extract new discriminative colors in the background, and compare them with  $C_f$  and  $C_b$ . The colors in  $C_f$  which are no more discriminative are removed, and new discriminative colors for the background are added to  $C_b$ .

**Updating the long-term model.** A bag of patches not only contains previously appeared patches, but also records their frequency, *i.e.*, their recurrence time. The bags of patches are updated independently only when their corresponding positions (*i.e.* the crosspoints in the grid) are totally visible. By that means, only the foreground are involved in model adaptation, thus avoiding model drift caused by the intervention of background regions. Once locating the model, the positions of the bags in the model are also determined. We compare the support (a  $K \times K$  square) of each bag with the foreground region. If the support of the bag is entirely inside the foreground, it is considered to be visible, and will be updated. Otherwise, it will be not updated at this time.

To update a bag of patches, we crop a new  $K \times K$  patch at the bag's position, and compared it with the maintained patches by calculating their SSD. If the cropped patch is very similar to a previously maintained patch, then the frequency of the maintained patch is increased by 1, otherwise the new patch is added to the list with initial frequency as 1. With such a simple but efficient model adaptation approach, the long-term local appearances of the target are effectively captured and preserved.

## 6 Experiments

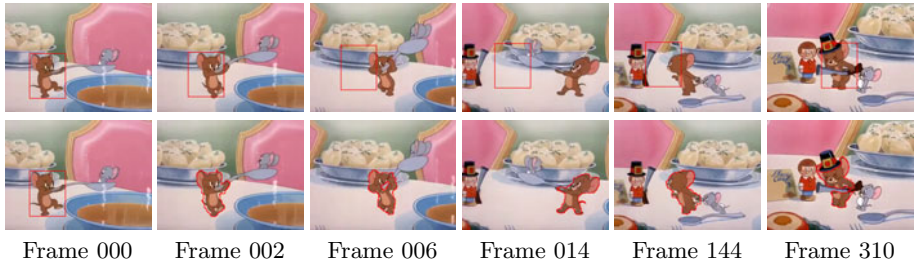
During tracking, if the size of the target is  $W \times H$ , then a surrounding region with size  $1.5W \times 1.5H$  is considered as its neighboring background, where salient points and discriminative color regions are extracted. We applied some morphological operators such as erosion and dilation to reduce the small noises in the scribbles. The computational cost of our approach is mostly ascribed to the matting algorithm. It is related to the amount of the pixels with uncertain alpha values before matting, which is generally dependent on the object size. In our method, much more scribbles are provided compared with user input, which makes matting faster. For example, our method can averagely process one frame per second in **Tom and Jerry** sequence without code optimization in our Matlab implementation, where the object size is around  $150 \times 150$ . This implies a great potential of a real-time implementation in C++. As a fast matting technique [17] has been proposed recently, the computational complexity is no longer a critical issue in our algorithm.

### 6.1 Comparison

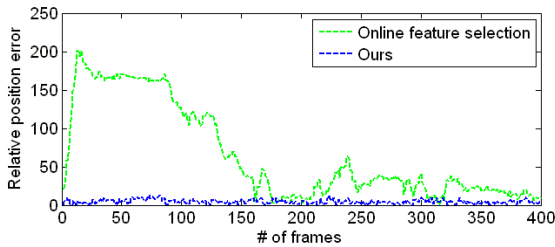
We first compared our method with Collins' method [1], in which they perform feature selection to discriminate the foreground and background. In the **Tom and**

---

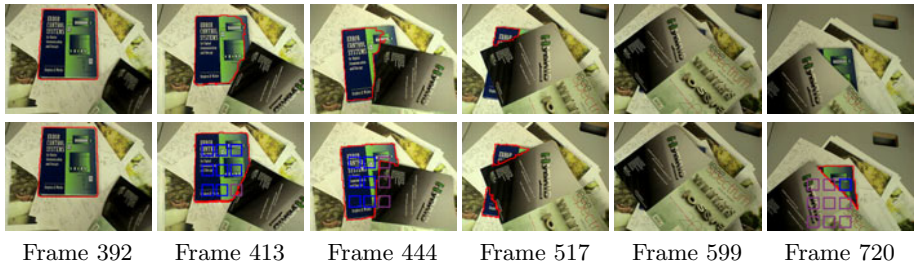
<sup>3</sup> Actually it is a simple clustering process, and the frequencies of the patches are the weights of the clusters.



**Fig. 6.** Comparison with Collins' online feature selection. Top: Tracking results by online feature selection. Bottom: Tracking results by our method.



**Fig. 7.** Quantitative Comparison with Collins' online feature selection



**Fig. 8.** Comparison with video matting. Top: Tracking results by video matting. Bottom: Tracking results by our method.

Jerry sequence, our approach can accurately obtain the boundary of Jerry, especially when he is holding a spoon or carrying a gun, while Collins' method drifted in the very beginning due to the fast motion, as we can see in Fig. 6. Notice that even we have a rough initialization at Frame 000 where some background is included, our method can still correctly get the boundary eventually. We also provided a quantitative comparison (Fig. 7). Our method shows a higher accuracy.

We also compared our method with video matting [9]. To make their method work in the tracking scenario (*i.e.* automatic processing), all the user input except for the first frame is removed. We both use the closed-form matting method [7] for fair comparison. As we can see in Fig. 8, in video matting the



Fig. 9. Tracking target with fast motion and large deformation

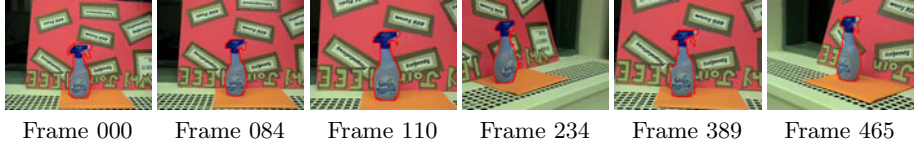


Fig. 10. Handling scale change and out-of-plane rotation

estimation of optical flow is not accurate at motion discontinuities and in homogeneous regions, therefore their cutout result is not satisfying. Furthermore, they cannot handle occlusion. By contrast, our method can always adaptively keep the boundary of the book. In this sequence, blue squares means that this bag is not occluded and will be updated, while purple squares means that this bag is currently under occlusion. At Frame 517, none of the bags is totally visible, the model therefore stops updating. At Frame 720, when the book reappears from complete occlusion, our model can successfully relocate it.

## 6.2 More Scenarios

We tested our method in more complex and challenging scenarios. The **skating** sequence has very fast motion and significant deformation. Motion blur can be clearly observed on each frame. The background keeps changing fast, especially when the skater is jumping. Given the initialization at Frame 000, Our method performs very well and gives a clear cutout for the skater, as shown in Fig. 9. Our method can also handle scaling and out-of-plane rotation in Fig. 10.

## 7 Conclusion

This paper introduces matting into a tracking framework and proposes a closed-loop model adaptation scheme. In our framework, the scribbles for matting are automatically generated by tracking, while matting results are used to obtain accurate boundaries of the object and to update the tracking model. Our work validates the applicability of automated matting in a tracking system, and meanwhile largely avoids the model drift problem in tracking with the aid of matting results. The proposed framework can be considered as a fundamental guideline on the combination of matting and tracking. In such a framework, each component in the closed loop can be further explored to improve the tracking performance.

**Acknowledgements.** This work was supported in part by National Science Foundation grant IIS-0347877, IIS-0916607, and US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504.

## References

1. Collins, R., Liu, Y., Leordeanu, M.: On-line selection of discriminative tracking features. *IEEE Trans. on PAMI* (2005)
2. Nguyen, H., Smeulders, A.: Robust tracking using foreground-background texture discrimination. *IJCV*, 277–293 (2006)
3. Babenko, B., Yang, M., Belongie, S.: Visual tracking with online multiple instance learning. In: *CVPR* (2009)
4. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You’ll never walk alone: modeling social behavior for multi-target tracking. In: *ICCV* (2009)
5. Yu, T., Wu, Y.: Collaborative tracking of multiple targets. In: *CVPR* (2004)
6. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. *IEEE Trans. on PAMI*, 810–815 (2006)
7. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. *IEEE trans. on PAMI*, 228–242 (2008)
8. Wang, J., Cohen, M.: Image and video matting: a survey. *Foundations and Trends in Computer Graphics and Vision*, 97–175 (2007)
9. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R.: Video matting of complex scenes. In: *SIGGRAPH* (2002)
10. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapCut: robust video object cutout using localized classifiers. In: *SIGGRAPH* (2009)
11. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: *IJCV* (2004)
12. Kuhl, F.P., Giardina, C.R.: Elliptic fourier features of a closed contour. *Computer Graphics and Image Processing* (1982)
13. Hager, G., Belhumeur, P.: Real-time tracking of image regions with changes in geometry and illumination. In: *CVPR* (1996)
14. Zhou, Y., Tao, H.: A background layer model for object tracking through occlusion. In: *ICCV* (2003)
15. Yin, Z., Collins, R.: Shape constrained figure-ground segmentation and tracking. In: *CVPR* (2009)
16. Ren, X., Malik, J.: Tracking as repeated figure/ground segmentation. In: *CVPR* (2007)
17. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: *CVPR* (2010)
18. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
19. Yang, M., Hua, G., Wu, Y.: Context-aware visual tracking. *IEEE Trans. on PAMI*, 1195–1209 (2009)
20. Wu, Y., Fan, J.: Contextual flow. In: *CVPR* (2009)
21. Avidan, S.: Ensemble tracking. In: *CVPR* (2005)
22. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: *CVPR* (2000)
23. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: *CVPR* (1998)

# Gaussian-Like Spatial Priors for Articulated Tracking

Søren Hauberg, Stefan Sommer, and Kim Steenstrup Pedersen

The eScience Centre, Dept. of Computer Science, University of Copenhagen  
{hauberg,sommer,kimstp}@diku.dk

**Abstract.** We present an analysis of the spatial covariance structure of an articulated motion prior in which joint angles have a known covariance structure. From this, a well-known, but often ignored, deficiency of the kinematic skeleton representation becomes clear: spatial variance not only depends on limb lengths, but also increases as the kinematic chains are traversed. We then present two similar Gaussian-like motion priors that are explicitly expressed spatially and as such avoids any variance coming from the representation. The resulting priors are both simple and easy to implement, yet they provide superior predictions.

**Keywords:** Articulated Tracking, Motion Analysis, Motion Priors, Spatial Priors, Statistics on Manifolds, Kinematic Skeletons.

## 1 Articulated Tracking

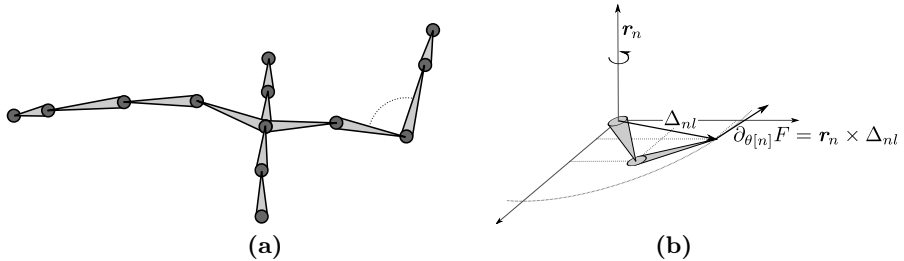
Three dimensional articulated human motion tracking is the process of estimating the configuration of body parts over time from sensor input [1]. One approach to this estimation is to use motion capture equipment where e.g. electromagnetic markers are attached to the body and then tracked in three dimensions. While this approach gives accurate results, it is intrusive and cannot be used outside laboratory settings. Alternatively, computer vision systems can be used for non-intrusive analysis. These systems usually perform some sort of optimisation for finding the best configuration of body parts. This optimisation is often guided by a system for predicting future motion. This paper concerns such a predictive system for general purpose tracking. Unlike most previous work, we build the actual predictive models in spatial coordinates, rather than working directly in the space of configuration parameters.

In the computer vision based scenario, the objective is to estimate the human pose in each image in a sequence. When only using a single camera, or a narrow baseline stereo camera, this is inherently difficult due to self-occlusions. This manifests itself in that the distribution of the human pose is multi-modal with an unknown number of modes. To reliably estimate this pose distribution we need methods that cope well with multi-modal distributions. Currently, the best method for such problems is the particle filter [2], which represents the distribution as a set of weighted samples. These samples are propagated in time using a predictive model and assigned a weight according to a data likelihood. As

such, the particle filter requires two subsystems: one for computing likelihoods by comparing the image data to a sample from the pose distribution, and one for predicting future poses. In terms of optimisation, the latter guides the search for the optimal pose. In practice, the predictive system is essential in making the particle filter computationally feasible, as it can drastically reduce the number of needed samples.

## 1.1 The Kinematic Skeleton

Before discussing the issues of human motion analysis, we pause to introduce the actual representation of the human pose. In this paper, we use the *kinematic skeleton* (see Fig. 1(a)), which is by far the most common choice [1]. This representation is a collection of connected rigid bones organised in a tree structure. Each bone can be rotated at the point of connection between the bone and its parent. We will refer to such a connection point as a *joint*.



**Fig. 1.** (a) A rendering of the kinematic skeleton. Each bone position is computed by a rotation and a translation relative to its parent. The circles, are collectively referred to as the *end-effectors*. (b) The derivative of an end point with respect to a joint angle. This is computed as the cross product of the rotational axis  $r_n$  and the vector from the joint to the end-effector.

We model the bones as having known constant length (i.e. rigid), so the direction of each bone constitute the only degrees of freedom in the kinematic skeleton. The direction in each joint can be parametrised with a vector of angles, noticing that different joints may have different number of degrees of freedom. We may collect all joint angle vectors into one large vector  $\theta$  representing all joint angles in the model. This vector will then be confined to the  $N$  dimensional torus  $\mathbb{T}^N$ .

**Forward Kinematics.** From known bone lengths and a joint angle vector  $\theta$ , it is straight-forward to compute the spatial coordinates of the bones. Specifically, the purpose is to compute the spatial coordinates of the end points of each bone. This process is started at the root of the tree structure and moves recursively along the branches, which are known as the *kinematic chains*.

The root of the tree is placed at the origin of the coordinate system. The end point of the next bone along a kinematic chain is then computed by rotating

the coordinate system and translating the root along a fixed axis relative to the parent bone, i.e.

$$\mathbf{a}_l = \mathbf{R}_l (\mathbf{a}_{l-1} + \mathbf{t}_l) \quad , \quad (1)$$

where  $\mathbf{a}_l$  is the  $l^{\text{th}}$  end point, and  $\mathbf{R}_l$  and  $\mathbf{t}_l$  denotes a rotation and a translation respectively. The rotation is parametrised by the relevant components of the pose vector  $\boldsymbol{\theta}$  and the length of the translation corresponds to the known length of the bone. We can repeat this process recursively until the entire kinematic tree has been traversed. This process is known as *Forward Kinematics* [3].

The rotation matrix  $\mathbf{R}_l$  of the  $l^{\text{th}}$  bone is parametrised by parts of  $\boldsymbol{\theta}$ . The actual number of used parameters depends on the specific joint. For elbow joints, we use one parameter, while we use three parameters to control all other joints. These two different joint types are respectively known as *hinge joints* and *ball joints*.

Using forward kinematics, we can compute the spatial coordinates of the end points of the individual bones. These are collectively referred to as *end-effectors*. In Fig. 1a these are drawn as circles. We will denote the coordinates of all end-effectors by  $F(\boldsymbol{\theta})$ . We will assume the skeleton contains  $L$  end-effectors, such that  $F(\boldsymbol{\theta}) \in \mathbb{R}^{3L}$ .

It should be clear that while  $F(\boldsymbol{\theta}) \in \mathbb{R}^{3L}$ , the end-effectors does not cover all of this space. There is, for instance, an upper bound on how far the hands can be apart. Specifically, we see that  $F(\boldsymbol{\theta}) \in \mathcal{M} \subset \mathbb{R}^{3L}$ , where  $\mathcal{M}$  is a compact differentiable manifold embedded in  $\mathbb{R}^{3L}$  (since  $\mathbb{T}^N$  is compact and  $F$  is an injective function with full-rank Jacobian).

**Derivative of Forward Kinematics.** Later, we shall be in need of the Jacobian of  $F$ . This consists of a column for each component of  $\boldsymbol{\theta}$ . Each such column can be computed in a straightforward manner [4]. Let  $\mathbf{r}_n$  denote the unit-length rotational axis of the  $n^{\text{th}}$  angle and  $\Delta_{nl}$  the vector from the joint to the  $l^{\text{th}}$  end-effector. The entries of the column corresponding to the  $l^{\text{th}}$  end-effector can then be computed as  $\partial_{\boldsymbol{\theta}_{[n]}} F_l = \mathbf{r}_n \times \Delta_{nl}$ . This is merely the tangent of the circle formed by the end-effector when rotating the joint in question as is illustrated in Fig. 1b.

**Joint Constraints.** In the human body, bones cannot move freely. A simple example is the elbow joint, which can approximately only bend between 0 and 120 degrees. To represent this,  $\boldsymbol{\theta}$  is confined to a subset  $\Theta$  of  $\mathbb{T}^N$ . With this further restriction,  $\mathcal{M}$  becomes a manifold with boundary.

For simplicity,  $\Theta$  is often defined by confining each component of  $\boldsymbol{\theta}$  to an interval, i.e.  $\Theta = \prod_{n=1}^N [l_n, u_n]$ , where  $l_n$  and  $u_n$  denote the lower and upper bounds of the  $n^{\text{th}}$  component. This type of constraints on the angles is often called *box constraints* [5].

## 1.2 Related Work

Most work in the articulated tracking literature falls in two categories. Either the focus is on improving the image likelihoods or on improving the predictions.



Due to space constraints, we forgo a review of various likelihood models as this paper is focused on prediction. For an overview of likelihood models, see the review paper by Poppe [1].

Most work on improving the predictions, is focused on learning motion specific priors, such as for *walking* [6, 7, 8, 9, 10, 11, 12]. Currently, the most popular approach is to restrict the tracker to some subspace of the joint angle space. Examples include, the work of Sidenbladh et al [10] where the motion is confined to a linear subspace which is learned using PCA. Similarly, Sminchisescu and Jepson [8] use spectral embedding to learn a non-linear subspace; Lu et al [9] use the Laplacian Eigenmaps Latent Variable Model [13] to perform the learning, and Urtasun et al [14] use a Scaled Gaussian Process Latent Variable Model [15]. This strategy has been improved even further by Urtasun et al [12] and Wang et al [7] such that a stochastic process is learned in the non-linear subspace as well. These approaches all seem to both stabilise the tracking and make it computationally less demanding. The downside is, of course, that the priors are only applicable when studying specific motions.

When it comes to general purpose priors, surprisingly little work has been done. Such priors are not only useful for studying general motion but can also be useful as hyperpriors for learning motion specific priors. The common understanding seems to be that the best general purpose prior is to assume that the joint angles follow a Gaussian distribution. Specifically, many researchers assume

$$p_{\text{angle}}(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}) \propto \mathcal{N}(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \Sigma_{\boldsymbol{\theta}}) \mathcal{U}_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_t) , \quad (2)$$

where  $\mathcal{U}_{\boldsymbol{\Theta}}$  denotes the uniform distribution on  $\boldsymbol{\Theta}$  enforcing the angular constraints and the subscript  $t$  denotes time. We shall call this model the *Angular Prior*. In practice,  $\Sigma_{\boldsymbol{\theta}}$  is often assumed to be diagonal or isotropic. This model has, amongst others, been applied by Sidenbladh et al [10], Balan et al [16] and Bandouch et al [17]. At first sight, this model seems quite innocent, but, as we shall see, it has a severe downside.

### 1.3 Our Contribution and Organisation of the Paper

In Sec. 2 we provide an analysis of the spatial covariance of the common motion prior from Eq. 2. While the formal analysis is novel, its conclusions are not surprising. In Sec. 3, we suggest two similar motion priors that are explicitly designed to avoid the problems identified in Sec. 2. This work constitutes the main technical contribution of the paper. In order to compare the priors we implement an articulated tracker, which requires a likelihood model. We briefly describe a simple model for this in Sec. 4. The resulting comparison between priors is performed in Sec. 5 and the paper is concluded in Sec. 6.

## 2 Spatial Covariance Structure of the Angular Prior

While the covariance structure of  $\boldsymbol{\theta}_t$  in Eq. 2 is straight-forward, the covariance of  $F(\boldsymbol{\theta}_t)$  is less simple. This is due to two phenomena:

1. **Variance depends on distance between joint and end-effector.** When a joint angle is changed, it alters the position of the end point of the limb attached to the joint. This end point is moved on a circle with radius corresponding to the distance between the joint and the end point. This means the end point of a limb far away from the joint can change drastically with small changes of the joint angle.
2. **Variance accumulates.** When a joint angle is changed, all limbs that are further down the kinematic chain will move. This means that when, e.g., the shoulder joint changes both hand and elbow moves. Since the hand also moves when the elbow joint changes, we see that the hand position varies more than the elbow position.

Neither of these two phenomena seem to have come from well-founded modelling perspectives.

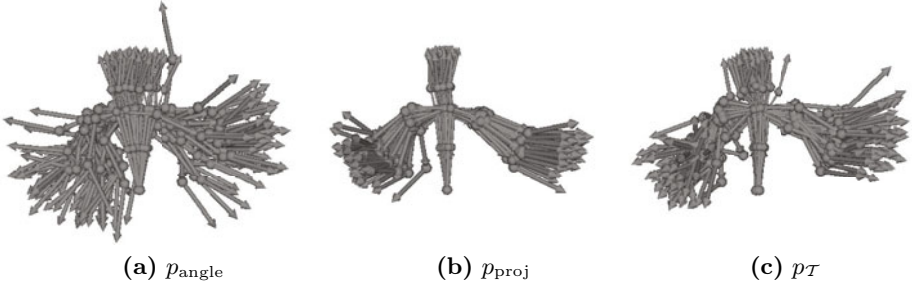
To get a better understanding of the covariance of limb positions, we seek an expression for  $\text{cov}[F(\boldsymbol{\theta}_t)]$ . Since  $F(\boldsymbol{\theta}_t)$  lies on a non-linear manifold  $\mathcal{M}$  in  $\mathbb{R}^{3L}$ , such an analysis is not straight-forward. Instead of computing the covariance on this manifold, we compute it in the tangent space at the mean value  $\bar{\boldsymbol{\theta}}_t = \mathbb{E}(\boldsymbol{\theta}_t)$  [18]. This requires the Logarithm map of  $\mathcal{M}$ , which we simply approximate by the Jacobian  $\mathbf{J}_{\bar{\boldsymbol{\theta}}_t} = \partial_{\boldsymbol{\theta}_t} F(\boldsymbol{\theta}_t)|_{\boldsymbol{\theta}_t=\bar{\boldsymbol{\theta}}_t}$  of the forward kinematics function, such that

$$\text{cov}[F(\boldsymbol{\theta}_t)] \approx \text{cov}[\mathbf{J}_{\bar{\boldsymbol{\theta}}_t} \boldsymbol{\theta}_t] = \mathbf{J}_{\bar{\boldsymbol{\theta}}_t} \text{cov}[\boldsymbol{\theta}_t] \mathbf{J}_{\bar{\boldsymbol{\theta}}_t}^T = \mathbf{J}_{\bar{\boldsymbol{\theta}}_t} \boldsymbol{\Sigma}_{\boldsymbol{\theta}_t} \mathbf{J}_{\bar{\boldsymbol{\theta}}_t}^T . \quad (3)$$

As can be seen, the covariance of the limb positions is highly dependent on the Jacobian of  $F$ . A slightly different interpretation of the used approximation is that we linearise  $F$  around the mean, and then compute the covariance.

We note that  $\|\partial_{\boldsymbol{\theta}_t[n]} F_l\| = \|\Delta_{nl}\|$ , meaning that the variance of a limb is linearly dependent on the distance between the joint and the limb end point. This is the first of the above mentioned phenomena. The second phenomena comes from the summation in the matrix product in Eq. 3. It should be stressed that this behaviour is a consequence of the choice of representation and will appear in any model that is expressed in terms of joint angles unless it explicitly performs some means of compensation. We feel this is unfortunate, as the behaviour does not seem to have its origins in an explicit model design decision. Specifically, it hardly seems to have any relationship with natural human motion (see the discussion of Fig. 2a below).

In practice, both of the above mentioned phenomena are highly visible in the model predictions. In Fig. 2a we show 50 samples from Eq. 2. Here, the joint angles are assumed to be independent, and the individual variances are learned from ground truth data of a sequence studied in Sec. 5. As can be seen the spatial variance increases as the kinematic chains are traversed. In practice, this behaviour reduces the predictive power of the model drastically; in our experience the model practically has no predictive power at all. Bandouch et al [17] suggested using *Partitioned Sampling* [19] to overcome this problem. This boils down to fitting individual limbs one at a time as the kinematic chains are traversed, such that e.g. the upper arm is fitted to the data before the lower



**Fig. 2.** Fifty samples from the different priors. The variance parameters for these distributions were assumed independent and was learned from ground truth data for a sequence studied in Sec. 5. (a) The angular prior  $p_{\text{angle}}$ . (b) The projected prior  $p_{\text{proj}}$ . (c) The tangent space prior  $p_{\mathcal{T}}$ .

arm. While this approach works, we believe it is better to fix the model rather than work around its limitations. As such, we suggest expressing the predictive model directly in terms of spatial limb positions.

### 3 Two Spatial Priors

Informally, we would like a prior where each limb position is following a Gaussian distribution, i.e.

$$p_{\text{idea}}(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}) = \mathcal{N}(F(\boldsymbol{\theta}_t) | F(\boldsymbol{\theta}_{t-1}), \Sigma) . \tag{4}$$

This is, however, *not* possible as the Gaussian distribution covers the entire  $\mathbb{R}^{3L}$ , whereas  $F(\boldsymbol{\theta}_t)$  is confined to  $\mathcal{M}$ . In the following, we suggest two ways of overcoming this problem.

#### 3.1 Projected Prior

The most straight-forward approach is to define  $p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$  by projecting Eq. 4 onto  $\mathcal{M}$ , i.e.

$$p_{\text{proj}}(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}) = \text{proj}_{\mathcal{M}}[\mathcal{N}(F(\boldsymbol{\theta}_t) | F(\boldsymbol{\theta}_{t-1}), \Sigma_{\text{proj}})] . \tag{5}$$

When using a particle filter for tracking, we only need to be able to draw samples from the prior model. We can easily do this by sampling from Eq. 4 and projecting the result onto  $\mathcal{M}$ . This, however, requires an algorithm for performing the projection.

Let  $\mathbf{x}_t$  denote a sample from Eq. 4; we now seek  $\hat{\boldsymbol{\theta}}_t$  such that  $F(\hat{\boldsymbol{\theta}}_t) = \text{proj}_{\mathcal{M}}[\mathbf{x}_t]$ . We perform the projection in a direct manor by seeking

$$\hat{\boldsymbol{\theta}}_t = \min_{\boldsymbol{\theta}_t} \|\mathbf{x}_t - F(\boldsymbol{\theta}_t)\|^2 \quad \text{s.t.} \quad \mathbf{l} \leq \boldsymbol{\theta}_t \leq \mathbf{u} , \tag{6}$$

where the constraints corresponds to the joint limits. This is an overdetermined constrained non-linear least-squares problem, that can be solved by any standard algorithm. We employ a projected steepest descent with line-search [5], where the search is started in  $\theta_{t-1}$ . To perform this optimisation, we need the gradient of Eq. 6, which is readily evaluated as  $\partial_{\theta_t} \|\mathbf{x}_t - F(\theta_t)\|^2 = 2(\mathbf{x}_t - F(\theta_t))^T \mathbf{J}_{\theta_t}$ .

In Fig. 2b we show 50 samples from this distribution, where  $\Sigma_{\text{proj}}$  is assumed to be a diagonal matrix with entries that have been learned from ground truth data of a sequence from Sec. 5. As can be seen, this prior is far less variant than the Gaussian prior  $p_{\text{angle}}$  on joint angles.

### 3.2 Tangent Space Prior

While the projected prior provides us with a suitable prior, it does come with the price of having to solve a non-linear least-squares problem. If the prior is to be used as e.g a regularisation term in a more complicated learning scheme, this can complicate the models substantially. As an alternative, we suggest a slight simplification that allows us to skip the non-linear optimisation. Instead of letting  $F(\theta_t)$  be Gaussian distributed in  $\mathbb{R}^{3L}$ , we define it as being Gaussian distributed in the tangent space  $\mathcal{T}$  of  $\mathcal{M}$  at  $F(\theta_{t-1})$ . That is, we define our prior such that

$$p_{\mathcal{T}}(\theta_t | \theta_{t-1}) = \mathcal{N}_{\mathcal{T}}(F(\theta_t) | F(\theta_{t-1}), \Sigma_{\mathcal{T}}) , \quad (7)$$

where  $\mathcal{N}_{\mathcal{T}}$  denotes a Gaussian distribution in  $\mathcal{T}$ . A basis of the tangent space is given by the columns of the Jacobian  $\mathbf{J}_{\theta_{t-1}}$ . From Eq. 3 we know that the covariance structure near  $F(\theta_{t-1})$  in this model is  $\Sigma_{\mathcal{T}} = \mathbf{J}_{\theta_{t-1}} \Sigma_{\theta} \mathbf{J}_{\theta_{t-1}}^T$ . In general,  $\mathbf{J}_{\theta_{t-1}}$  is not square, so we cannot isolate  $\Sigma_{\theta}$  from this equation simply by inverting  $\mathbf{J}_{\theta_{t-1}}$ . Instead, we take the straight-forward route and use the pseudoinverse of  $\mathbf{J}_{\theta_{t-1}}$ , such that

$$p_{\text{tang}}(\theta_t | \theta_{t-1}) \propto \mathcal{N} \left( \theta_t | \theta_{t-1}, \mathbf{J}_{\theta_{t-1}}^{\dagger} \Sigma_{\mathcal{T}} (\mathbf{J}_{\theta_{t-1}}^{\dagger})^T \right) \mathcal{U}_{\Theta}(\theta_t) , \quad (8)$$

where  $\mathbf{J}_{\theta_{t-1}}^{\dagger} = (\mathbf{J}_{\theta_{t-1}}^T \mathbf{J}_{\theta_{t-1}})^{-1} \mathbf{J}_{\theta_{t-1}}^T$  denotes the pseudoinverse of  $\mathbf{J}_{\theta_{t-1}}$ . If we consider  $\mathbf{J}_{\theta_{t-1}}$  a function from  $\mathbb{T}^N$  to  $\mathcal{T}$  then  $\mathbf{J}_{\theta_{t-1}}^{\dagger}$  is indeed the inverse of this function. One interpretation of this prior is that it is the normal distribution in angle space that provides the best linear approximation of a given normal distribution in the spatial domain.

To sample from this distribution, we generate a sample  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathcal{T}})$ . This is then moved into the joint angle space by letting  $\theta_t = (\mathbf{J}_{\theta_{t-1}}^{\dagger})^T \mathbf{x} + \theta_{t-1}$ . In order to respect joint limits, we truncate joint values that exceeds their limitations. This simple scheme works well in practice.

In Fig. 2c we show 50 samples from this distribution, where  $\Sigma_{\mathcal{T}}$  is the same as the projected prior in Fig. 2b. As can be seen, this prior behaves somewhat more variant than  $p_{\text{proj}}$ , but far less than  $p_{\text{angle}}$ .

## 4 Visual Measurements

To actually implement an articulated tracker, we need a system for making visual measurements, i.e. a likelihood model. To keep the paper focused on prediction, we use a simple likelihood model based on a consumer stereo camera<sup>1</sup>. This camera provides a dense set of three dimensional points  $\mathbf{Z} = \{z_1, \dots, z_K\}$  in each frame. The objective of the likelihood model then becomes to measure how well a pose hypothesis matches the points. We assume that each point is independent and that the distance between a point and the skin of the human follows a zero-mean Gaussian distribution, i.e.

$$p(\mathbf{Z}|\boldsymbol{\theta}_t) \propto \prod_{k=1}^K \exp\left(-\frac{D^2(\boldsymbol{\theta}_t, z_k)}{2\sigma^2}\right), \quad (9)$$

where  $D^2(\boldsymbol{\theta}_t, z_k)$  denotes the square distance between the point  $z_k$  and the skin of the pose  $\boldsymbol{\theta}_t$ . To make the model robust with respect to outliers in the data we threshold the distance function  $D$  such that it never exceeds a given threshold.

We also need to define the skin of a pose, such that we can compute distances between this and a data point. Here, we define the skin of a bone as a cylinder with main axis corresponding to the bone itself. Since we only have a single view point, we discard the half of the cylinder that is not visible. The skin of the entire pose is then defined as the union of these half-cylinders. The distance between a point and this skin can then be computed as the smallest distance from the point to any of the half-cylinders.

## 5 Experimental Results

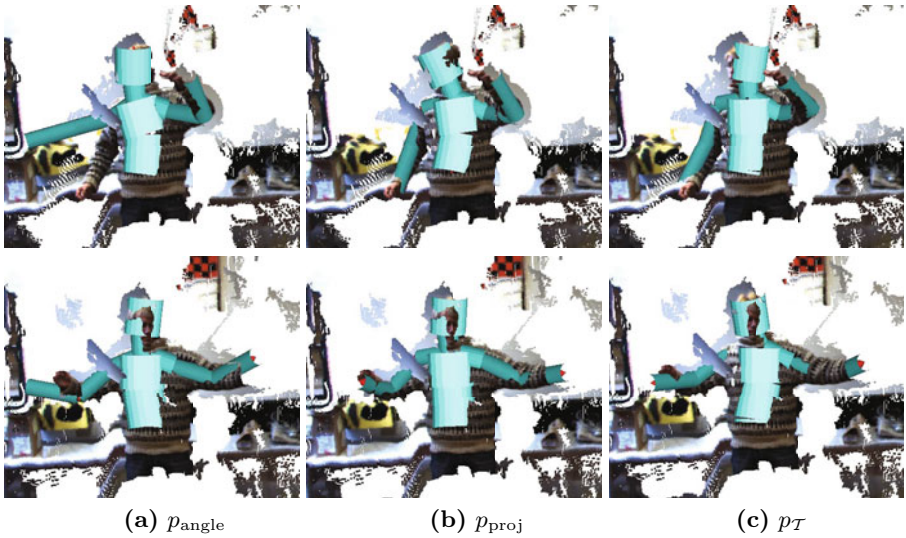
To build an articulated tracker we combine the likelihood model with the suggested priors using a particle filter. This provides us with a set of weighted samples from which we estimate the current pose as the weighted average.

We seek to compare the three suggested priors,  $p_{\text{angle}}$ ,  $p_{\text{proj}}$  and  $p_{\text{tang}}$ . As the base of our comparison, we estimate the pose in each frame of a sequence using a particle filter with 10.000 samples, which is plenty to provide a good estimate. This will then serve as our ground truth data. As we are studying a general purpose motion model, we assume that each prior has a diagonal covariance structure. These variances are then learned from the ground truth data to give each prior the best possible working conditions.

We apply the three prior models to a sequence where a person is standing in place and mostly moving his arms. We vary the number of particles in the three tracking systems between 25 and 1500. The results are available as videos on-line<sup>2</sup> and some selected frames are available in Fig. 3. The general tendency is that the projected prior provides the most accurate and smooth results for a given number of particles. Next, we seek to quantify this observation.

<sup>1</sup> <http://www.ptgrey.com/products/bumblebee2/>

<sup>2</sup> <http://humim.org/eccv2010/>



**Fig. 3.** Results attained using 150 and 250 samples superimposed on the image data. Top row is using 150 particles, while bottom row is using 250 particles. (a) Using the angular prior. (b) Using the projected prior. (c) Using the tangent space prior.

To compare the attained results to the ground truth data, we apply a simple spatial error measure [20, 116]. This measures the average distance between limb end points in the attained results and the ground truth data. This measure is then averaged across frames, such that the error measure becomes

$$\mathcal{E} = \frac{1}{TL} \sum_{t=1}^T \sum_{l=1}^L \|\mathbf{a}_{lt} - \mathbf{a}'_{lt}\|, \quad (10)$$

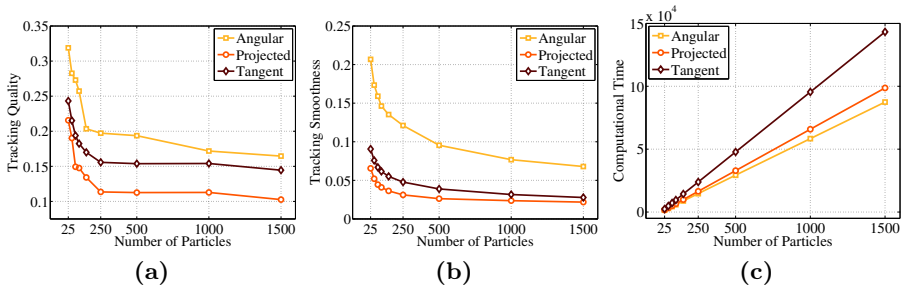
where  $\mathbf{a}_{lt}$  is the spatial end point of the  $l^{\text{th}}$  limb at time  $t$  in the attained results, and  $\mathbf{a}'_{lt}$  is the same point in the ground truth data. This measure is reported for the different priors in Fig. 4a. As can be seen, the projected prior is consistently better than the tangent space prior, which in turn is consistently better than the angular prior. One explanation of why the projected prior outperforms the tangent space prior could be that  $\mathcal{M}$  has substantial curvature. This explanation is also in tune with the findings of Sommer et.al [21].

If the observation density  $p(\mathbf{Z}_t|\boldsymbol{\theta}_t)$  is noisy, the motion model acts as a smoothing filter. This can be of particular importance when observations are missing, e.g. during self-occlusions. Thus, when evaluating the quality of a motion model it can be helpful to look at the smoothness of the attained pose sequence. To measure this, we simply compute the average size of the temporal gradient. We approximate this gradient using finite differences, and hence use

$$S = \frac{1}{TL} \sum_{t=1}^T \sum_{l=1}^L \|\mathbf{a}_{lt} - \mathbf{a}_{l,t-1}\| \quad (11)$$

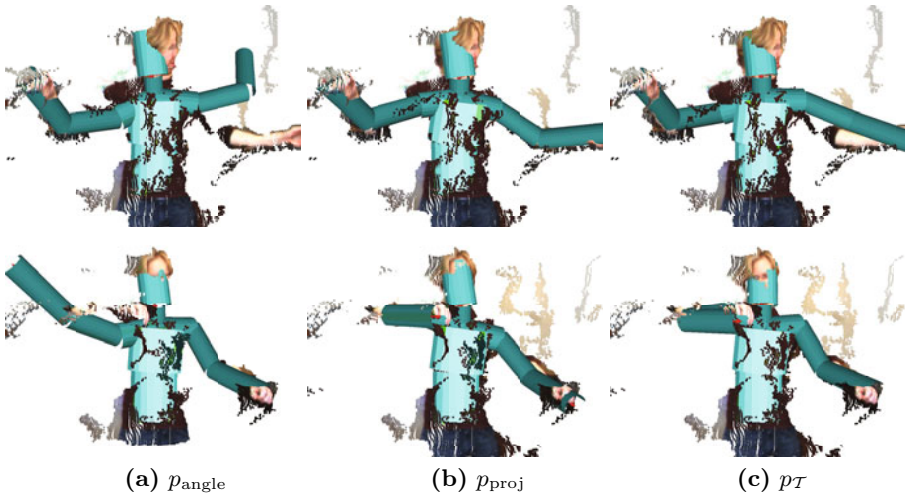
as a measure of smoothness. This is reported in Fig. 4b. It can be seen that the projected prior and the tangent space prior give pose sequences that are almost equally smooth; both being consistently much more smooth than the angular prior. This is also quite visible in the on-line videos.

So far we have seen that both suggested priors outperform the angular prior in terms of quality. The suggested priors are, however, computationally more demanding. One should therefore ask if it is computationally less expensive to simply increase the number of particles while using the angular prior. In Fig. 4c we report the running time of the tracking systems using the different priors. As can be seen, the projected prior is only slightly more expensive than the angular prior, whereas the tangent space prior is somewhat more expensive than the two other models. The latter result is somewhat surprising given the simplicity of the tangent space prior; we believe that this is caused by choices of numerical methods. In practice both of the suggested priors give better results than the angular prior at a fixed amount of computational resources, where the projected prior is consistently the best.

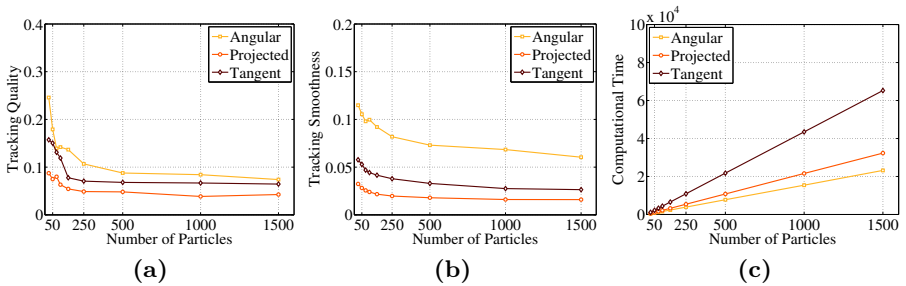


**Fig. 4.** Performance of the three priors. All reported numbers are averaged over several trials. (a) The error measure  $\mathcal{E}$  as a function of the number of particles. The average standard deviation of  $\mathcal{E}$  with respect to the trials are 0.018 for the angular prior, 0.008 for the projected prior and 0.009 for the tangent space prior. (b) The smoothness measure  $\mathcal{S}$  as a function of the number of particles. The average standard deviation of  $\mathcal{S}$  with respect to the trials are 0.0028 for the angular prior, 0.0009 for the projected prior and 0.0016 for the tangent space prior. (c) The computational time as a function of the number of particles.

We now repeat the experiment for a second sequence, using the same parameters as before. In Fig. 5 we show the tracking results in selected frames for the three discussed priors. As before, videos are available on-line<sup>2</sup>. Essentially, we make the same observations as before: the projected prior provides the best and most smooth results, followed by the tangent space prior with the angular prior consistently giving the worst results. This can also be seen in Fig. 6 where the error and smoothness measures are plotted along with the running time of the methods. Again, we see that for a given amount of computational resources, the projected prior consistently provides the best results.



**Fig. 5.** Results attained using 150 and 250 samples superimposed on the image data. Top row is using 150 particles, while bottom row is using 250 particles. (a) Using the angular prior. (b) Using the projected prior. (c) Using the tangent space prior.



**Fig. 6.** Performance of the three priors. All reported numbers are averaged over several trials. (a) The error measure  $\mathcal{E}$  as a function of the number of particles. The average standard deviation of  $\mathcal{E}$  with respect to the trials are 0.021 for the angular prior, 0.007 for the projected prior and 0.015 for the tangent space prior. (b) The smoothness measure  $\mathcal{S}$  as a function of the number of particles. The average standard deviation of  $\mathcal{S}$  with respect to the trials are 0.002 for the angular prior, 0.0004 for the projected prior and 0.001 for the tangent space prior. (c) The computational time as a function of the number of particles.

## 6 Discussion

We have presented an analysis of the commonly used prior which assumes Gaussian distributed joint angles, and have shown that this behaves less than desirable spatially. Specifically, we have analysed the covariance of this prior in the tangent space of the pose manifold. This has clearly illustrated that small changes in a



joint angle can lead to large spatial changes. Since this instability is ill-suited for predicting articulated motion, we have suggested to define the prior directly in spatial coordinates.

Since human motion is restricted to a manifold  $\mathcal{M} \subset \mathbb{R}^{3L}$ , we, however, need to define the prior in this domain. We have suggested two means of accomplishing this goal. One builds the prior by projecting onto the manifold and one builds the prior in the tangent space of the manifold. Both solutions have shown to outperform the ordinary angular prior in terms of both speed and accuracy. Of the two suggested priors, the projected prior seems to outperform the tangent space prior, both in terms of speed and quality. The tangent space prior does, however, have the advantage of simply being a normal distribution in joint angle space, which can make it more suitable as a prior when learning a motion specific model.

One advantage with building motion models spatially is that we can express motion specific knowledge quite simply. As an example, one can model a person standing in place simply by reducing the variance of the persons feet. This type of knowledge is non-trivial to include in models expressed in terms of joint angles.

The suggested priors can be interpreted as computationally efficient approximations of a Brownian motion on  $\mathcal{M}$ . We therefore find it interesting to investigate this connection further along with similar stochastic process models restricted to manifolds. In the future, we will also use the suggested priors as building blocks in more sophisticated motion specific models.

## References

1. Poppe, R.: Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding* 108, 4–18 (2007)
2. Cappé, O., Godsill, S.J., Moulines, E.: An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE* 95, 899–924 (2007)
3. Erleben, K., Sporring, J., Henriksen, K., Dohlmann, H.: *Physics Based Animation*. Charles River Media (2005)
4. Zhao, J., Badler, N.I.: Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transaction on Graphics* 13, 313–336 (1994)
5. Nocedal, J., Wright, S.J.: *Numerical optimization*. Springer Series in Operations Research. Springer, Heidelberg (1999)
6. Brubaker, M.A., Fleet, D.J., Hertzmann, A.: Physics-based person tracking using the anthropomorphic walker. *Int. J. of Comp. Vis.* 87, 140–155 (2010)
7. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian Process Dynamical Models for Human Motion. *Pattern Analysis and Machine Intelligence* 30, 283–298 (2008)
8. Sminchisescu, C., Jepson, A.: Generative modeling for continuous non-linearly embedded visual inference. In: *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 759–766. ACM, New York (2004)
9. Lu, Z., Carreira-Perpinan, M., Sminchisescu, C.: People Tracking with the Laplacian Eigenmaps Latent Variable Model. In: Platt, J., et al. (eds.) *Advances in Neural Inf. Proc. Systems*, vol. 20, pp. 1705–1712. MIT Press, Cambridge (2008)
10. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3d human figures using 2d image motion. In: Vernon, D. (ed.) *ECCV 2000*. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000)

11. Elgammal, A.M., Lee, C.S.: Tracking People on a Torus. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 31, 520–538 (2009)
12. Urtasun, R., Fleet, D.J., Fua, P.: 3D People Tracking with Gaussian Process Dynamical Models. In: *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 238–245 (2006)
13. Carreira-Perpinan, M.A., Lu, Z.: The Laplacian Eigenmaps Latent Variable Model. *JMLR W&P* 2, 59–66 (2007)
14. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: *Int. Conf. on Comp. Vis.*, vol. 1, pp. 403–410 (2005)
15. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. *ACM Transaction on Graphics* 23, 522–531 (2004)
16. Balan, A.O., Sigal, L., Black, M.J.: A quantitative evaluation of video-based 3d person tracking. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 349–356 (2005)
17. Bandouch, J., Engstler, F., Beetz, M.: Accurate human motion capture using an ergonomics-based anthropometric human model. In: Perales, F.J., Fisher, R.B. (eds.) *AMDO 2008. LNCS*, vol. 5098, pp. 248–258. Springer, Heidelberg (2008)
18. Pennec, X.: Probabilities and statistics on riemannian manifolds: Basic tools for geometric measurements. In: *NSIP*, pp. 194–198 (1999)
19. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. In: Vernon, D. (ed.) *ECCV 2000. LNCS*, vol. 1843, pp. 3–19. Springer, Heidelberg (2000)
20. Sigal, L., Bhatia, S., Roth, S., Black, M.J., Isard, M.: Tracking loose-limbed people. In: *CVPR '04: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 421–428 (2004)
21. Sommer, S., Lauze, F., Hauberg, S., Nielsen, M.: Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part VI. LNCS*, vol. 6316, pp. 43–56. Springer, Heidelberg (2010)

# Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow\*

Narayanan Sundaram, Thomas Brox, and Kurt Keutzer

University of California at Berkeley  
{narayans,brox,keutzer}@eecs.berkeley.edu

**Abstract.** Dense and accurate motion tracking is an important requirement for many video feature extraction algorithms. In this paper we provide a method for computing point trajectories based on a fast parallel implementation of a recent optical flow algorithm that tolerates fast motion. The parallel implementation of large displacement optical flow runs about  $78\times$  faster than the serial C++ version. This makes it practical to use in a variety of applications, among them point tracking. In the course of obtaining the fast implementation, we also proved that the fixed point matrix obtained in the optical flow technique is positive semi-definite. We compare the point tracking to the most commonly used motion tracker - the KLT tracker - on a number of sequences with ground truth motion. Our resulting technique tracks up to three orders of magnitude more points and is 46% more accurate than the KLT tracker. It also provides a tracking density of 48% and has an occlusion error of 3% compared to a density of 0.1% and occlusion error of 8% for the KLT tracker. Compared to the Particle Video tracker, we achieve 66% better accuracy while retaining the ability to handle large displacements while running an order of magnitude faster.

## 1 Introduction

When analyzing video data, motion is probably the most important cue, and the most common techniques to exploit this information are difference images, optical flow, and point tracking. Since difference images restrict us to static cameras and we want to extract rich and unrestricted motion information, we will focus only on the last two techniques. The goal here is to enable accurate motion tracking for a large set of points in the video in close to real time and in this paper, we make substantial progress towards that goal. The quality of both the estimated flow field and the set of point trajectories are very important as small differences in the quality of the input features can make a high level approach succeed or fail. To ensure accuracy, many methods only track a sparse set of points; however, dense motion tracking enables us to extract information at a much finer granularity compared to sparse feature correspondences. Hence, one

---

\* This work was supported by the German Academic Exchange Service (DAAD) and the Gigascale Systems Research Center, one of five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program.

wants to use the most recent motion estimation technique providing the most reliable motion features for a specific task. For dense and accurate tracking there are usually computational restrictions. Video data processing requires far more resources than the analysis of static images, as the amount of raw input data is significantly larger. This often hinders the use of high-quality motion estimation methods, which are usually quite slow [1] and require expensive computer clusters to run experiments efficiently. For this reason, ways to significantly speedup such methods on commodity hardware are an important contribution as they enable more efficient research in fields that build upon motion features. This is important as more processing is usually required to utilize this motion information for use in video processing applications. Fast implementations of the KLT tracker and optical flow [2,3] are examples that have certainly pushed research.

In this paper we present a fast GPU implementation of large displacement optical flow (LDOF) [4], a recent variational optical flow method that can deal with faster motion than previous optical flow techniques[5]. The numerical schemes used in [4] and most variational methods are based on a coarse-to-fine warping scheme, where each level provides an update by solving a nonlinear system given by the Euler-Lagrange equations followed by fixed point iterations and a linear solver, as described in [5]. However, the relaxation techniques used in the linear solver that work best for serial processors are not efficient on parallel processors. We investigate alternative solvers that run well on parallel hardware, in particular, red-black relaxations and the conjugate gradient method. We show that the conjugate gradient method is faster than red-black relaxations, especially on larger images. We also prove that the fixed point matrix is positive semi-definite, thus guaranteeing the convergence of the conjugate gradient algorithm. We obtain a speedup of about  $78\times$ , which allows us to compute high quality LDOF for  $640\times 480$  images in 1.8 seconds. Extrapolating the current progress in GPU technology, the same code will even run in real-time in only a few years. While additional speedups are often obtained at the cost of lower quality, we ensured in our implementation that the quality of the original method is preserved.

We also propose a method for dense point tracking by building upon the fast implementation of large displacement optical flow. Point trajectories are needed whenever an approach builds on long term motion analysis. The dominant method used for this task is the KLT tracker [6], which is a sparse technique that only tracks a very small number of designated feature points. While for many tasks like camera calibration such sparse point trajectories are totally sufficient, other tasks like motion segmentation or structure-from-motion would potentially benefit from higher densities. In [1] and [7], a method for point tracking based on dense variational optical flow has been suggested. The method proposed in [1] is computationally very expensive and impractical to use on large datasets without acceleration. The point tracking we propose uses a similar technique, as points are propagated by means of the optical flow field; however, we do not build upon another energy minimization procedure that detects occluded points mainly by appearance, but do the occlusion reasoning by

---

<sup>1</sup> Executables and mex functions can be found at the authors' websites.

a forward-backward consistency check of the optical flow. In a quantitative comparison on some sequences from [8], where close to ground truth optical flow has been established by manually annotating the objects in the scene, we show that we can establish much denser point trajectories with better quality than the KLT tracker. At the same time, our method is more accurate and runs an order of magnitude faster than the technique in [7]. Such fast, high quality tracking will enable new applications such as better video summarization or activity recognition through improved tracking of limbs and balls in sports videos.

## 2 Related Work

Finding efficient solutions to variational optical flow problems has been an active area of research. On serial hardware, multi-grid solvers based on Gauss-Seidel have been proposed in [9]. A GPU implementation of the formulation in [9] has been proposed using Jacobi solvers [10]. Compared to [10], our implementation handles large displacements through dense descriptor matching. Such extensions enable us to handle fast motion well [11], [4]. A multi-grid red-black relaxation has been suggested in a parallel implementation of the linear CLG method [12]. Very efficient GPU implementations of other variational optical flow models have been proposed in [3], [13], [14].

The conjugate gradient algorithm is a popular solver for convex problems and has been used for optical flow problems with convex quadratic optimization [15]. In order to theoretically justify the use of conjugate gradients, we prove that the system matrix of general variational optical flow methods is positive semi-definite and thus the conjugate gradient solver is guaranteed to converge. It was previously proven that the Horn-Schunck matrix is positive definite [16]. Our proof is more general and applicable to most variational formulations [9], [5], [17] and [11].

The most popular point tracker is the Kanade-Lucas-Tomasi (KLT) tracker [6], which constructs an image pyramid, chooses points that have sufficient structure and tracks them across frames. New features are periodically detected to make up for the loss of features because of occlusions and tracking errors. This is generally considered to be fast and accurate, but it tracks only a few points. Efficient GPU implementations of the KLT tracker have been released in [18] and [2]. While the KLT algorithm itself is quite old, the implementation in [2] compensates for changes in camera exposure to make it more robust. Non-local point trackers that use global information have also been proposed [19].

The more advanced point tracker in [1] and [7] tracks points by building on top of a variational technique. This comes with high computational costs. It takes more than 100 seconds to track points between a pair of  $720 \times 480$  frames. Moreover, this technique cannot deal with large displacements of small structures like limbs, and it has never been shown whether tracking based on variational flow actually performs better than the classic KLT tracker.

### 3 Large Displacement Optical Flow on the GPU

Large displacement optical flow (LDOF) is a variational technique that integrates discrete point matches, namely the midpoints of regions, into the continuous energy formulation and optimizes this energy by a coarse-to-fine scheme to estimate large displacements also for small scale structures [11]. As pointed out in [4], region matching can be replaced with matching other features like densely sampled histograms of oriented gradients (HOG) [20]. These simpler features allow us to implement both the variational solver and the discrete matching efficiently on the GPU.

The considered energy functional that is minimized reads:

$$\begin{aligned}
 E(\mathbf{w}) = & \int_{\Omega} \Psi_1(|I_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - I_1(\mathbf{x})|^2) + \gamma \Psi_2(|\nabla I_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - \nabla I_1(\mathbf{x})|^2) d\mathbf{x} \\
 & + \beta \int_{\Omega} \delta(\mathbf{x}) \rho(\mathbf{x}) \Psi_3(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2) d\mathbf{x} + \int_{\Omega} \delta(\mathbf{x}) |\mathbf{f}_2(\mathbf{x} + \mathbf{w}_1(\mathbf{x})) - \mathbf{f}_1(\mathbf{x})|^2 d\mathbf{x} \quad (1) \\
 & + \alpha \int_{\Omega} \Psi_S(|\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2) d\mathbf{x}
 \end{aligned}$$

where  $\mathbf{w} = (u \ v)^T$  and  $\Psi_*(s^2)$  is a general penalizer function with its derivative  $\Psi'_*(s^2) > 0$ . A popular choice in the literature is  $\Psi_*(s^2) = \sqrt{s^2 + \epsilon^2}$  [4].

Since speed and accuracy are foremost in solving the optical flow problem, it is necessary to take advantage of the improvements in modern microprocessors to aid the solution. In particular, parallelism has emerged as a key to performance scaling. Hence, it is necessary to study and develop algorithms and techniques that best utilize multiple processing elements simultaneously.

A parallel implementation of the descriptor matching is relatively straightforward since several points are being searched for in parallel without any dependencies between them. It is important, however, to take advantage of coalesced memory accesses (vector loads/stores) in order to maximize the performance of the GPU. In the rest of the section, we will focus on the parallel implementation of the variational solver that considers these point correspondences.

#### 3.1 Variational Solver on the GPU

We minimize (1) by writing the Euler-Lagrange equations and solving them through a coarse-to-fine scheme with fixed point iterations. This results in a sequence of linear systems to be solved, where each pixel corresponds to two coupled equations in the linear system:

$$\begin{aligned}
 (\Psi'_1 I_x^{k2} + \gamma \Psi'_2 (I_{xx}^{k2} + I_{xy}^{k2}) + \beta \rho \Psi'_3) du^{k,l+1} + (\Psi'_2 I_x^k I_y^k + \gamma \Psi'_2 (I_{xx}^k I_{xy}^k + I_{xy}^k I_{yy}^k)) dv^{k,l+1} \\
 - \alpha \operatorname{div}(\Psi'_S \nabla (u^k + du^{k,l+1})) = -\Psi'_1 (I_x^k I_z^k) + \gamma \Psi'_2 (I_{xx}^k I_{xz}^k + I_{xy}^k I_{yz}^k) - \beta \rho \Psi'_3 (u^k - u_1) \quad (2)
 \end{aligned}$$

$$\begin{aligned}
 (\Psi'_1 I_y^{k2} + \gamma \Psi'_2 (I_{yy}^{k2} + I_{xy}^{k2}) + \beta \rho \Psi'_3) dv^{k,l+1} + (\Psi'_2 I_x^k I_y^k + \gamma \Psi'_2 (I_{xx}^k I_{xy}^k + I_{xy}^k I_{yy}^k)) du^{k,l+1} \\
 - \alpha \operatorname{div}(\Psi'_S \nabla (v^k + dv^{k,l+1})) = -\Psi'_1 (I_y^k I_z^k) + \gamma \Psi'_2 (I_{yy}^k I_{yz}^k + I_{xy}^k I_{xz}^k) - \beta \rho \Psi'_3 (v^k - v_1)
 \end{aligned}$$

For details on the derivation of these equation we refer to [4]. From symmetry considerations, the discretization usually produces a symmetric block pentadiagonal matrix with  $2 \times 2$  blocks (for a 5-point Laplacian stencil). From equation (2), it is clear that only the diagonal blocks are dense, while the off-diagonal blocks are diagonal matrices. In fact, for the isotropic functionals we consider here, they are scaled identity matrices.

**Positive semi-definiteness of the fixed point matrix.** We have proven in [21] that the fixed point matrix is symmetric positive semi-definite because (a) the diagonal blocks are positive definite and (b) the matrix is block diagonally dominant [22]. The detailed proof is provided in [21]. An interesting takeaway from the proof is that it is not restricted to convex penalty functions  $\Psi_*$ . The only restriction on  $\Psi_*$  is that it should be increasing. Moreover, the proof technique generalizes to most variational optical flow methods, e.g. [5], [9], [11] and [17].

**Linear solvers.** On the CPU, the linear system is usually solved using *Gauss-Seidel* relaxations, which have been empirically shown to be very efficient in this setting [23]. The Gauss-Seidel method is guaranteed to converge if the matrix is symmetric positive definite. Unfortunately, the Gauss-Seidel technique is inherently sequential as it updates the points in a serial fashion. It is hard to parallelize it efficiently on multi-core machines and even harder on GPUs.

It is possible to choose relaxation methods that have slightly worse convergence characteristics, but are easy to parallelize, such as *Red-black relaxation* [24]. A single red-black relaxation consists of two half iterations - each half iteration updates every alternate point (called red and black points). The updates to all the red points are inherently parallel as all the dependencies for updating a red point are the neighboring black pixels and vice versa. Usually, this method is used with successive overrelaxation. Since we have a set of coupled equations, each relaxation will update  $(u_i, v_i)$  using a  $2 \times 2$  matrix solve. Red-black relaxations have been used in a previous parallel optical flow solver [12].

Besides red-black relaxation, we consider the *Conjugate gradient* method. This requires symmetric positive definiteness as a necessary and sufficient condition for convergence. The convergence of the conjugate gradient technique depends heavily on the condition number of the matrix  $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$ . The condition numbers of the matrices obtained in the optical flow problems are very large and hence, convergence is usually slow.

A standard technique for improving convergence for ill-conditioned matrices is preconditioning to reduce the condition number of the system matrix. The preconditioner must be symmetric and positive definite. The special structure of the matrix allows for several regular pre-conditioners that work well in practice. In particular, we know that the diagonal blocks of the matrix are positive definite. Hence, a block diagonal matrix with only the diagonal blocks of the matrix is symmetric and positive definite and forms a good pre-conditioner. This preconditioner is usually referred to as a *block Jacobi preconditioner*. From now on, unless specified, we use the term conjugate gradient solver to refer to the preconditioned conjugate gradient solver with a block Jacobi preconditioner.



**Fig. 1. Left: (a)** Initial points in the first frame using a fixed subsampling grid. **Middle: (b)** Frame number 15 **Right: (c)** Frame number 30 of the cameramotion sequence. Figure best viewed in color.

Performing this algorithmic exploration is important as choosing the right algorithm for the right platform is essential for getting the best speed-accuracy tradeoff. This fast LDOF implementation can now be used to track points in video.

## 4 Point Tracking with Large Displacement Optical Flow

We demonstrate the utility of our LDOF implementation by suggesting a point tracker. In contrast to traditional local point trackers, like KLT [6], variational optical flow takes global smoothness constraints into account. This allows the tracking of far more points as the flow field is dense and tracking is not restricted to a few feature points. Moreover, large displacement optical flow enables tracking limbs or other fast objects more reliably than conventional trackers.

Our tracking algorithm works as follows: a set of points is initialized in the first frame of a video. In principle, we can initialize with every pixel, as the flow field is dense. However, areas without any structure are problematic for tracking with variational optical flow as well. For this reason, we remove points that do not show any structure in their vicinity as measured by the second eigenvalue  $\lambda_2$  of the structure tensor

$$J_\rho = K_\rho * \sum_{k=1}^3 \nabla I_k \nabla I_k^\top, \quad (3)$$

where  $K_\rho$  is a Gaussian kernel with standard deviation  $\rho = 1$ . We ignore all points where  $\lambda_2$  is smaller than a certain portion of the average  $\lambda_2$  in the image.

Depending on the application, one may actually be interested in fewer tracks that uniformly cover the image domain. This can be achieved by spatially subsampling the initial points. Fig. 1 shows a subsampling by factor 8. The coverage of the image is still much denser than with usual keypoint trackers.

Each of the points can be tracked to the next frame by using the optical flow field  $\mathbf{w} := (u, v)^\top$ :

$$(x_{t+1}, y_{t+1})^\top = (x_t, y_t)^\top + (u_t(x_t, y_t), v_t(x_t, y_t))^\top. \quad (4)$$

As the optical flow is subpixel accurate,  $x$  and  $y$  will usually end up between grid points. We use bilinear interpolation to infer the flow at these points.



The tracking has to be stopped as soon as a point gets occluded. This is extremely important, otherwise the point will share the motion of two differently moving objects. Usually occlusion is detected by comparing the appearance of points over time. In contrast, we detect occlusions by checking the consistency of the forward and the backward flow, which we found to be much more reliable. In a non-occlusion case, the backward flow vector points in the inverse direction as the forward flow vector:  $u_t(x_t, y_t) = -\hat{u}_t(x_t + u_t, y_t + v_t)$  and  $v_t(x_t, y_t) = -\hat{v}_t(x_t + u_t, y_t + v_t)$ , where  $\hat{\mathbf{w}}_t := (\hat{u}_t, \hat{v}_t)$  denotes the flow from frame  $t + 1$  back to frame  $t$ . If this consistency requirement is not satisfied, the point is either getting occluded at  $t + 1$  or the flow was not correctly estimated. Both are good reasons to stop tracking this point at  $t$ . Since there are always some small estimation errors in the optical flow, we grant a tolerance interval that allows estimation errors to increase linearly with the motion magnitude:

$$|\mathbf{w} + \hat{\mathbf{w}}|^2 < 0.01 (|\mathbf{w}|^2 + |\hat{\mathbf{w}}|^2) + 0.5. \quad (5)$$

We also stop tracking points on motion boundaries. The exact location of the motion boundary, as estimated by the optical flow, fluctuates a little. This can lead to the same effect as with occlusions: a tracked point drifts to the other side of the boundary and partially shares the motion of two different objects. To avoid this effect we stop tracking a point if

$$|\nabla u|^2 + |\nabla v|^2 > 0.01 |\mathbf{w}|^2 + 0.002. \quad (6)$$

In order to fill the empty areas caused by disocclusion or scaling, in each new frame we initialize new tracks in unoccupied areas using the same strategy as for the first frame.

## 5 Results

The implementation platform consists of an Intel Core2 Quad Q9550 processor running at 2.83GHz in conjunction with a Nvidia GTX 480 GPU. For the LDOF implementations almost all of the computation is done on the GPU and only minimal amount of data is transferred between the CPU and the GPU. We use Nvidia CUDA tools (v3.0) for programming the GPU. The CPU implementation is a well written hand coded serial C++ code that was vectorized using the Intel compiler with all the optimizations enabled.

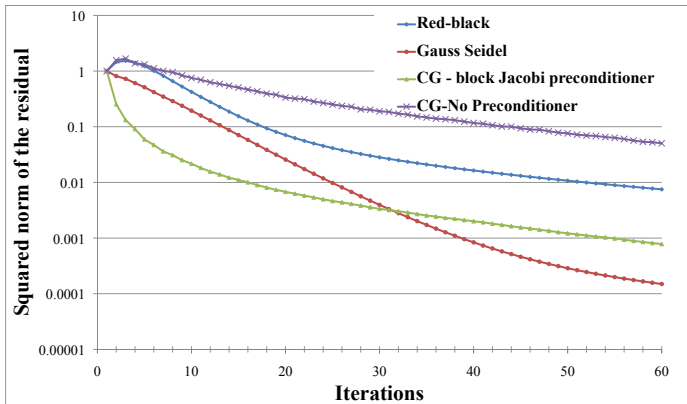
For the tracking experiments, the KLT tracker used also runs on GPUs. A description of the algorithm is provided in [2]. The implementation in [2] also compensates for changes in camera exposure and provides real-time performance on the GPU considered. Default parameters were used unless otherwise specified.

### 5.1 GPU Accelerated Large Displacement Optical Flow

Runtime for large displacement optical flow has come down from 143 seconds for the previous serial implementation on CPU to 1.84 seconds for the parallel

implementation on GPU, a speedup of  $78\times$  for an image size of  $640 \times 480$ . This implementation searches for HOG matches in a neighborhood of  $\pm 80$  pixels, uses  $\eta = 0.95$ , 5 fixed point iterations and 10 Conjugate gradient iterations to achieve the same overall AAE as the CPU version on the Middlebury dataset. It is also possible to run the optical flow algorithm at a slightly reduced accuracy (AAE increase of about 11%) at 1.1 seconds per frame. The performance of the linear solver is critical to the overall application runtime. Hence we look closely at the choice of the linear solver that enabled this speedup.

**Performance of linear solvers.** Figure 2 shows the convergence of different solvers for the optical flow problem. We measure convergence through the squared norm of the residual  $\|b - Ax^m\|^2$ . The rates of convergence are derived from 8 different matrices from images in the Middlebury dataset [25]. Red-black and Gauss-Seidel solvers use successive overrelaxation with  $\omega = 1.85$ . The matrices considered were of the largest scale (smaller scales show very similar results). The initial vector in all the methods was an all-zero vector. Using a better initialization procedure (the result of a previous fixed point iteration, for instance) also shows similar results.

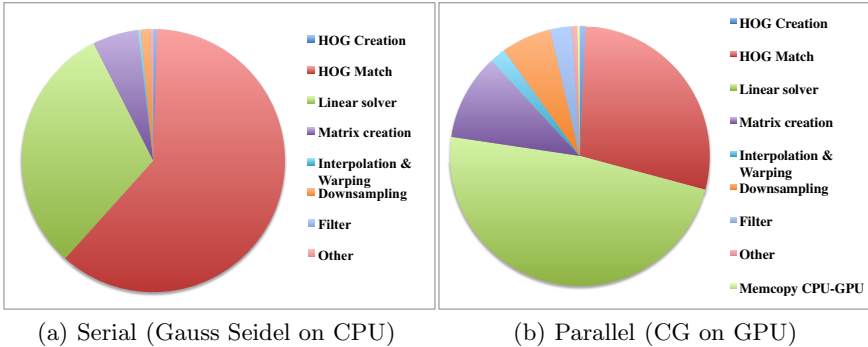


**Fig. 2.** Rates of convergence for different techniques considered. Y-axis shows the value of the residual normalized to the initial residual value averaged over 8 different matrices. Figure best viewed in color.

From Fig. 2 we can see why the Gauss-Seidel solver is the preferred choice for serial platforms. It converges well and is relatively simple to implement. In the numerical scheme at hand, however, we do not desire absolute convergence, as solving any one linear system completely is not important to the solution of the nonlinear system. It is more important to have a quick way of refining the solution and removing all the large errors. For a few iterations (30 or less), it is clear that the preconditioned conjugate gradient solver converges fastest. Non-preconditioned conjugate gradient is not as efficient because of the high condition number of the matrix.

**Table 1.** Average time taken by the linear solvers for achieving residual norm  $< 10^{-2}$ 

Linear Solver	Time taken (in milliseconds)
Gauss-Seidel	395.13
Red-black	11.97
Conjugate Gradient	8.39

**Fig. 3.** Breakdown of execution times for serial and parallel variational optical flow solvers. Both solvers are run at a scale factor of 0.95, with 5 fixed point iterations and 25 Gauss-Seidel iterations/10 CG iterations to achieve similar AAE. Figure best viewed in color.

Although it is clear from Fig. 2 that conjugate gradient converges quickly in terms of the number of iterations required, a single iteration of conjugate gradient requires more computation than a Gauss-Seidel or a red-black iteration. Table 1 shows the runtimes of the solvers. Even though red-black relaxations are also parallel, we can see from Fig. 2 that we require roughly  $3\times$  as many red-black iterations as conjugate gradient iterations to achieve the same accuracy. Red-black iterations are  $1.4\times$  slower than CG overall. Gauss-Seidel iterations, running on the CPU, are  $47\times$  slower compared to conjugate gradient on the GPU.

Figure 3 shows the breakdown of the serial optical flow solver that uses Gauss-Seidel and the parallel solver that uses conjugate gradient. The solvers were run with  $\eta = 0.95$ , 5 fixed point iterations and 25 Gauss-Seidel iterations/10 Conjugate gradient iterations to achieve similar AAE on the Middlebury dataset. From both Figure 3(a) and 3(b), it is clear that the HOG matching and the linear solver are the most computation intensive components in the solvers. In both cases, they take more than 75% of the total runtime.

The major bottleneck in the conjugate gradient solver is the sparse matrix-vector multiply (SpMV). In order to optimize SpMV, the sparse matrix was laid out in memory as a set of images each corresponding to a particular non-zero diagonal. This, along with several other optimizations (caching in local scratchpad memory, avoiding control overheads, ensuring vector loads/stores) enables the SpMV to run at 53 GFlops on the GPU. This is significant considering that

**Table 2.** Average Angular Error (in degrees) for images in the Middlebury dataset

Data	Dimetrodon	Grove2	Grove3	Hydrangea	RubberWhale	Urban2	Urban3	Venus	Average
AAE(CPU)	1.84	2.67	6.35	2.44	3.96	2.55	4.79	6.46	3.88
AAE(GPU)	1.84	2.51	5.94	2.37	3.91	2.47	5.43	6.38	3.86

the matrix is quite sparse ( $\leq 6$  non-zeros per row). Under such conditions, most of the time in the kernel is spent fetching data to and from GPU main memory. Similar behavior is seen with the red-black relaxations, where 25% of the time is spent in floating point operations, while 75% of the time is spent in memory loads and stores. Red-black relaxations also have less computation to communication ratio (all the points are read, but only half the points are updated), which reduces their performance.

**Accuracy.** Table 2 shows the average angular error measured using our technique on the Middlebury dataset. These results have been achieved with the setting ( $\gamma = 4$ ,  $\beta=30$ ,  $\alpha = 9$ ,  $\eta = 0.95$ , fixed point iterations = 5, Gauss-Seidel iterations = 25/CG iterations = 10). The data shows that the method provides similar accuracy to the CPU version while running fast on the GPU.

For faster computations, we use the parameter set ( $\eta = 0.75$ , 5 fixed point iterations, 10 linear solve iterations) to reduce the runtime by 38% with a degradation in AAE of 11%.

## 5.2 Tracking

We measure the accuracy of the tracking algorithms with the MIT sequences [8]. This dataset provides the ground truth optical flow for whole sequences and the sequences are much longer. This allows us to evaluate the accuracy of tracking algorithms. After obtaining the point trajectories from both KLT and LDOF, we track points using the given ground truth to predict their final destination. Tracking error is measured as the mean Euclidean distance between the final tracked position and the predicted position on the final frame according to the ground truth for all the tracked points. LDOF is run with  $\eta = 0.95$ , 5 fixed point iterations and 10 iterations for the linear solver in all the following experiments. Since the default parameters for the KLT tracker failed in tracking points in long sequences, we increased the threshold for positive identification of a feature from 1000 to 10000 (SSD-threshold parameter).

**Accuracy.** We compare the accuracy of the trackers for the entire length of the sequences. Since tracking algorithms should ideally track points over long times without losing points, we only consider those points that are tracked through all the frames. Trackers like KLT keep losing features and need to be constantly detecting new features every few frames to track well. From Table 3, it is clear that LDOF tracks almost three orders of magnitude more points than KLT with 46% improvement in overall accuracy. For tracking only the common points, the LDOF tracker is 32% better than KLT. These numbers exclude the fish sequence since it has transparent motion caused by dust particles moving in the water.

**Table 3.** Tracking accuracy of LDOF and KLT over the MIT sequences

Sequence name	Number of frames	All tracked points				Common points only			
		LDOF		KLT		LDOF		KLT	
		Mean error in pixels	Points tracked	Mean error in pixels	Points tracked	Mean error in pixels	Points tracked	Mean error in pixels	Points tracked
table	13	1.48	114651	3.78	363	1.04	293	1.39	293
camera	37	1.41	101662	3.78	278	1.01	185	2.64	185
fish	75	3.39	75907	35.62	106	3.12	53	5.9	53
hand	48	2.14	151018	3.11	480	1.87	429	2.39	429
toy	18	2.24	376701	2.88	866	1.70	712	1.89	712

**Table 4.** Tracking accuracy of LDOF and the Particle Video tracker over the 20 sequences used in [7]

Average number of frames	All tracked points				Common points only			
	LDOF		Particle Video		LDOF		Particle Video	
	Mean error in pixels	Points tracked	Mean error in pixels	Points tracked	Mean error in pixels	Points tracked	Mean error in pixels	Points tracked
61.5	0.83	109579	3.20	8967	0.84	3304	2.51	3304

**Table 5.** Occlusion handling by KLT and LDOF trackers based on region annotation from the MIT data set. Occluded tracks indicate tracks that are occluded according to the ground truth data, but not identified as such by the trackers.

Sequence	KLT		LDOF		
	Number of occluded tracks	Mean error with no occlusion	Number of occluded tracks	Mean error with no occlusion	Tracking Density (%)
table	11	2.73	853	1.41	39.6
camera	8	3.68	558	1.37	39.9
fish	30	31.79	8321	2.7	53.0
hand	10	2.90	2127	1.81	46.8
toy	31	2.58	5482	2.11	61.4

Although we were able to track this sequence well, performance on this sequence is sensitive to parameter changes.

Compared to the Particle Video point tracker in [7], our tracker is 66% more accurate for the common tracks. Since ground truth data does not exist for the sequences used in [7], it is not possible to have objective comparisons on metrics other than the average round trip error (The videos are mirrored temporally, so all unoccluded pixels should return to their starting positions). For comparison, we use only the full-length particle trajectories provided by the authors of [7] at <http://rvsn.csail.mit.edu/pv/data/pv>. The average statistics of both trackers over all the 20 sequences used in [7] are given in Table 4. More details on the comparison can be found in [21].

**Occlusion handling.** We use the region annotation data from the MIT dataset to measure the occlusion handling capabilities of the algorithms. The LDOF

**Table 6.** Tracking accuracy of LDOF and KLT for large displacements in the tennis sequence with manually marked correspondences. Numbers in parentheses indicate the number of annotated points that were tracked.

Frames	LDOF		KLT	
	Mean error in pixels	Points tracked on player	Mean error in pixels	Points tracked on player
490-495	2.55 (22)	8157	3.21 (19)	21
490-500	2.62 (8)	3690	4.12 (4)	4



**Fig. 4.** (Top) Frame 490 of the tennis sequence with (left) actual image, (middle) KLT points and (right) LDOF points. (Bottom) Frame 495 of the sequence with (left) actual image, (middle) KLT points and (right) LDOF points. Only points on the player are marked. KLT tracker points are marked larger for easy visual detection. Figure best viewed in color.

tracker has an occlusion error of 3% (tracks that drift between regions/objects) while the KLT tracker has an occlusion error of 8%. Given that KLT tracker is already very sparse, this amounts to a significant number of tracks that are not reliable (they do not reside on the same object for the entire time). After excluding all the tracks that were known to have occlusion errors, LDOF outperforms KLT by 44%. Since all the ground truth occlusions are known, we measure the tracking density (% of unoccluded points that the tracker was successful in tracking through the entire sequence without any occlusion errors). The LDOF tracker has an average tracking density of 48%, i.e., it tracks roughly half of the points that are not occluded for the entire length of the sequence, while KLT has a density of about 0.1%. Table 5 presents the data on occlusion handling.

**Large displacements.** The MIT sequences still mostly contain small displacements and hence KLT is able to track them well (if it does not lose the features); however, there are motion sequences with large displacements that are difficult for a tracker like KLT to capture. In the tennis sequence [11], there are frames

where the tennis player moves very fast producing motion that is hard to capture through simple optical flow techniques. Since ground truth data does not exist for this sequence we manually labeled the correspondences for 39 points on the player between frames 490, 495 and 500<sup>2</sup>. These points were feature points identified by KLT in frame 490. The results for the points tracked on the player are shown in Table 6 and Figure 4. It is clear that the LDOF tracker tracks more points with better accuracy, while capturing the large displacement of the leg.

**Runtime.** The cameramotion sequence with 37 frames of size  $640 \times 480$ , requires 135 seconds. Out of this, 125 seconds were spent on LDOF (both forward and backward flow). Such runtimes allow for convenient processing of large video sequences on a single machine equipped with cheap GPU hardware.

## 6 Conclusion

Fast, accurate and dense motion tracking is possible with large displacement optical flow (LDOF). We have provided a parallel version of LDOF that achieves a speedup of  $78 \times$  over the serial version. This has been possible through algorithmic exploration for the numerical solvers and an efficient parallel implementation of the large displacement optical flow algorithm on highly parallel processors (GPUs). Moreover, we have proposed a dense point tracker based on this fast LDOF implementation. Our experiments quantitatively show for the first time that tracking with dense motion estimation techniques provides better accuracy than KLT feature point tracking by 46% on long sequences and better occlusion handling. We also achieve 66% better accuracy than the Particle Video tracker. Our point tracker based on LDOF improves the density by up to three orders of magnitude compared to KLT and handles large displacements well, thus making it practical for use in a wide variety of video applications such as activity recognition or summarization in sports videos which require improved tracking of fast moving objects like balls and limbs.

## References

1. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision* 80, 72–91 (2008)
2. Zach, C., Gallup, D., Frahm, J.M.: Fast gain-adaptive KLT tracking on the GPU. In: *CVPR Workshop on Visual Computer Vision on GPU's, CVGPU* (2008)
3. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) *DAGM 2007*. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)
4. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (to appear, 2010)
5. 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)

<sup>2</sup> The manually labeled correspondence data can be found on the authors' website.

6. Shi, J., Tomasi, C.: Good features to track. In: CVPR, pp. 593–600 (1994)
7. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. In: CVPR (2006)
8. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. In: CVPR (2008)
9. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In: ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Washington, DC, USA, vol. 1, pp. 749–755. IEEE Computer Society, Los Alamitos (2005)
10. Grossauer, H., Thoman, P.: GPU-based multigrid: Real-time performance in high resolution nonlinear image processing. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 141–150. Springer, Heidelberg (2008)
11. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: CVPR (2009)
12. Gwosdek, P., Bruhn, A., Weickert, J.: High performance parallel optical flow algorithms on the Sony Playstation 3. *Vision, Modeling and Visualization*, 253–262 (2008)
13. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for TV-L1 optical flow. In: *Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar, Dagstuhl Castle, Germany, July 13-18*, pp. 23–45 (2009), Revised Papers
14. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: *Proc. of the British Machine Vision Conference, BMVC (2009)*
15. Lai, S.H., Vemuri, B.C.: Reliable and efficient computation of optical flow. *International Journal of Computer Vision* 29 (1998)
16. Mitiche, A., Mansouri, A.R.: On convergence of the Horn and Schunck optical-flow estimation method. *IEEE Transactions on Image Processing* 13, 848–852 (2004)
17. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vision* 61, 211–231 (2005)
18. Sinha, S.N., Frahm, J.M., Pollefeys, M., Genc, Y.: Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications* (2007)
19. Birchfield, S.T., Pundlik, S.J.: Joint tracking of features and edges. In: CVPR (2008)
20. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2006)
21. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by GPU-accelerated large displacement optical flow. Technical Report UCB/EECS-2010-104, EECS Department, University of California, Berkeley (2010)
22. Feingold, D.G., Varga, R.S.: Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem. *Pacific J. Math.* 12, 1241–1250 (1962)
23. Bruhn, A.: Variational Optic Flow Computation: Accurate Modelling and Efficient Numerics. PhD thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany (2006)
24. Stüben, K., Trottenberg, U.: Multigrid methods: Fundamental algorithms, model problem analysis and applications. *Lecture Notes in Mathematics*, vol. 960. Springer, Heidelberg (1982)
25. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV (2007)



# Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings

Stefano Pellegrini<sup>1</sup>, Andreas Ess<sup>1</sup>, and Luc Van Gool<sup>1,2</sup>

<sup>1</sup> Computer Vision Laboratory  
ETH Zurich

<sup>2</sup> ESAT-PSI / IBBT  
KU Leuven

{stefpell,aess,vangool}@vision.ee.ethz.ch

**Abstract.** We consider the problem of data association in a multi-person tracking context. In semi-crowded environments, people are still discernible as individually moving entities, that undergo many interactions with other people in their direct surrounding. Finding the correct association is therefore difficult, but higher-order social factors, such as group membership, are expected to ease the problem. However, estimating group membership is a chicken-and-egg problem: knowing pedestrian trajectories, it is rather easy to find out possible groupings in the data, but in crowded scenes, it is often difficult to estimate closely interacting trajectories without further knowledge about groups. To this end, we propose a third-order graphical model that is able to jointly estimate correct trajectories and group memberships over a short time window. A set of experiments on challenging data underline the importance of joint reasoning for data association in crowded scenarios.

**Keywords:** Grouping, Tracking, Data Association, Social Interaction.

## 1 Introduction

Tracking algorithms are an indispensable prerequisite for many higher-level computer vision tasks, ranging from surveillance to animation to automotive applications. Advances in observation models, such as object detectors or classification-based appearance models, have enabled tracking in previously infeasible scenarios. Still, tracking remains a challenging problem, especially in crowded environments. Tracking high numbers of pedestrians in such cases is even hard for humans. Usually, a manual annotator has to rely on higher-level reasoning, such as temporal information (that can go into the future) or social factors. Recent advances in the literature suggest that especially the latter can improve tracking performance. Typically employed social factors include a pedestrian's *destination*, *desired speed*, and *repulsion* from other individuals. Another factor is grouping behavior, which so far however has been largely ignored. For one, this is due to the fact that the grouping information (do two persons belong to the same group?) is not easily available. Still, groups constitute an

important part of a pedestrian's motion. As we will show in this paper, people behave differently when walking in groups as opposed to alone: when alone, they tend to keep a certain distance from others, passing by closely only if necessary, but mostly at different speeds. When in groups, they try to stay close enough with other members, walking at the same speed.

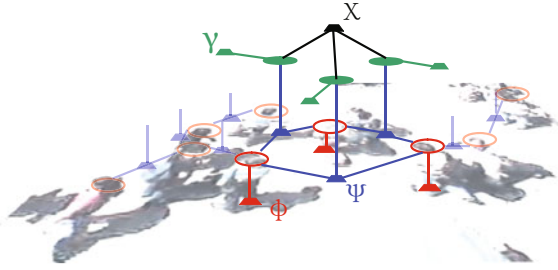
In this paper, we therefore aim at exploiting the interaction between different people for data association in a principled way. In particular, we model group relations and study their effect on trajectory prediction. The grouping between pedestrians is treated as a latent variable, which is estimated jointly together with the trajectory information. Our model of choice is a third-order CRF, with nodes in the lower level corresponding to pedestrians, connected by third-order links that represent possible groupings. Recent advances in discrete optimization provide powerful tools for carrying out (approximate) inference in such models.

In order to take advantage of as much information as possible, we adopt a hypothesize-and-verify strategy over a frame-based tracking approach. By operating on short time windows (typically, in the order of a few seconds), useful statistics over pedestrians and group participation can be obtained, while only introducing a small lag as opposed to global trackers. The proposed framework operates in two steps: first, it generates possible trajectory hypotheses for each person within the given time window, then it selects the best hypothesis, taking into account social factors, while at the same time estimating group membership.

The paper is organized as follows: Related work is explored in Section 2. In Section 3, our model for the joint estimation of trajectories and groupings using social factors is introduced. The training of the model by natural statistics of interaction is discussed in Section 4. The inference method is then presented in Section 5. Finally, we show experimental results in Section 6, before concluding the paper in Section 7.

## 2 Related Work

*Social behavior modeling.* Based on a variety of psychological, physical, and social factors, people tend to keep certain distances from each other when interacting. Already investigated in the 1960ies by Hall [1] as proxemics, these factors are meanwhile also used for the modeling of pedestrian motion. Applications include simulation [2–4], computer graphics [5, 6], and, in the last few years, also Computer Vision [7–14]. While all of these works include various social factors, grouping information was mostly ignored. Helbing *et al.*, in their seminal paper on the Social Force Model [2], include an attraction potential to model the group interaction. However, even in simulation applications, the notion of groups is rarely employed. In this paper, we will focus on group relations between subjects in a tracking setting, showing how to jointly estimate a person's correct trajectory and his group membership status. To the best of our knowledge, this paper is the first to explore the joint estimation of pedestrian trajectories and the grouping relations.



**Fig. 1.** Assumed higher-order model for joint trajectory and group finding (see text)

*Tracking.* Fostered by recent progress in object detection, there is an impressive body of work in single-person tracking-by-detection [15–20]. All propose different ways of handling the data association problem, but do not take advantage of any social factors beyond spatial exclusion principles. Only some researchers use social structures to improve tracking, most notably in crowded scenarios [7], or by modeling of collision avoidance-behavior [8, 11].

In this work, we focus on improving one building block of tracking—the data association—by taking advantage of social factors in a principled fashion. The proposed algorithm infers the best trajectory choice for each tracked object in a short time window. This thus means some latency as opposed to typical on-line trackers [15, 18, 19]. The method however does not need the entire time window either, as global approaches [17, 20, 21] model simple interactions of targets in an MCMC framework, but not accounting for groups. [16] also use a hypothesize-and-verify strategy, however, they do not model any social factors, and a hypothesis contains an entire person’s past, as opposed to a small window only. By operating in a shorter temporal window, our algorithm can take into account many more hypotheses, which is a requirement for tracking in challenging scenarios.

### 3 Group CRF

To improve data association in crowded scenarios, we want to jointly estimate pedestrian trajectories and their group relations. Fig. 1 shows the factor graph for the third-order CRF model we assume for this problem.

Given a starting frame, each tracking target  $i$  ( $i = 1 \dots N$ ) is modeled as a variable node (red empty circle, Fig. 1), where each possible state corresponds to the choice of one local trajectory hypothesis  $\mathbf{h}_i^m \in \mathcal{H}_i = \{\mathbf{h}_i^m\}_{m=1 \dots M_i}$ , with  $\mathcal{H}_i$  the set of hypotheses for one person. As a trajectory hypothesis  $\mathbf{h}_i^m$ , we consider a single subject’s possible future within a short time window. A joint assignment of hypotheses to all the subjects is defined as  $\mathbf{H}^q = [\mathbf{h}_1^{q(1)} \dots \mathbf{h}_N^{q(N)}]$ , where  $q$  is an assignment function that assigns each target  $i$  one hypothesis in  $\mathcal{H}_i$ .

<sup>1</sup> To reduce notational clutter, we drop the superscripts for  $\mathbf{h}$  and for  $\mathbf{H}$  in the following.

The group variable  $l_{c(ij)}$  (green filled circle, Fig. 1) indicates the group relation among the subjects  $i$  and  $j$ ,

$$l_{c(ij)} = \begin{cases} 1 & \text{if subject } i \text{ and } j \text{ belong to the same group} \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

with  $c(ij)$  an index function.

Two subjects  $i$  and  $j$  and the group variable  $l_{c(ij)}$  are linked by a factor of order three (blue factor in Fig. 1). This link variable is essential to take advantage of grouping relations in our model. Grouping is an equivalence relation, i.e. it fulfills reflexivity, symmetry, and transitivity. While reflexivity and symmetry are enforced by the graph construction, the transitivity constraint is encoded in a third-order factor (black factor in Fig. 1): given three subjects  $i, j$ , and  $k$  for which there exist the link variables  $l_{c(ij)}$ ,  $l_{c(ik)}$ , and  $l_{c(kj)}$ :

$$(l_{c(ij)} \wedge l_{c(ik)}) \rightarrow l_{c(kj)}. \quad (2)$$

The log-probability of a set of trajectories  $\mathbf{H}$  and a set of grouping relations  $\mathbf{L}$ , given an image  $\mathbf{I}$  and parameters  $\Theta$ , is given by

$$\begin{aligned} \log P(\mathbf{H}, \mathbf{L} | \mathbf{I}, \Theta) = & \sum_i \phi_i^{motion}(\mathbf{h}_i | \Theta_{\phi^{motion}}) + \sum_i \phi_i^{app}(\mathbf{h}_i | \mathbf{I}, \Theta_{\phi^{app}}) + \\ & \sum_{c(ij)} \gamma_{c(ij)}(l_{c(ij)} | \Theta_{\gamma}) + \sum_{ijc(ij)} \psi_{ijl_{c(ij)}}^{pos}(\mathbf{h}_i, \mathbf{h}_j, l_{c(ij)} | \Theta_{\psi^{pos}}) + \\ & \sum_{ijc(ij)} \psi_{ijl_{c(ij)}}^{ang}(\mathbf{h}_i, \mathbf{h}_j, l_{c(ij)} | \Theta_{\psi^{ang}}) + \\ & \sum_{c(ij)c(ik)c(kj)} \chi_{c(ij)c(ik)c(kj)}(l_{c(ij)}, l_{c(ik)}, l_{c(kj)} | \Theta_{\chi}) - \log Z(I, \Theta), \end{aligned} \quad (3)$$

where  $\phi^{app}$  and  $\phi^{motion}$  model, respectively, the appearance and motion of a trajectory,  $\gamma_{c(ij)}$  models the prior over a relation being of type group or not,  $\psi_{ijl_{c(ij)}}^{pos}$ ,  $\psi_{ijl_{c(ij)}}^{ang}$  model the grouping relation and  $Z(I, \Theta)$  is the usual partition function making sure that the probability density function sums to one.

## 4 Learning the Parameters

Learning the parameters of the model in Eq. 3 could be done by maximizing the conditional likelihood of the parameters given the data. However, this is hard because of the partition function  $Z$ . Instead, inspired by piecewise training [22], we learn simple statistics from the data and define the terms in the Eq. 3 as a combination of these statistics. In particular we overparametrize the trajectory  $\mathbf{h}_i$  as a sequence  $[\mathbf{p}_i^0, s_i^0, \alpha_i^0 \dots \mathbf{p}_i^{T-1}, s_i^{T-1}, \alpha_i^{T-1}]$  of, respectively, position, speed and orientation and extract simple statistics over these terms, rather than over the whole trajectory. In doing so, we use a non-parametric approach, by building histograms to estimate densities. The parameters  $\Theta$  can be interpreted as the



**Fig. 2.** A snapshot from the sequence used in this paper. We only use data inside the red bounding box, to avoid stairs on the left and too heavy shadows in the upper part.

entries of these histograms. To reduce the notational clutter we will drop in the following the dependence on  $\Theta$ .

In the analysis of the data, we make, when appropriate, a distinction between people walking and people standing. Besides believing that these two classes can have different statistics indeed, we are motivated for making this distinction by a technical limitation: the orientation estimate is hard and unreliable for standing people, while it can be approximated by the direction of motion for moving people. We therefore choose an empirical threshold of  $0.15\text{m/s}$ <sup>2</sup> to distinguish between the two modes.

In the following, we will show the relevant statistics that we used in our model.

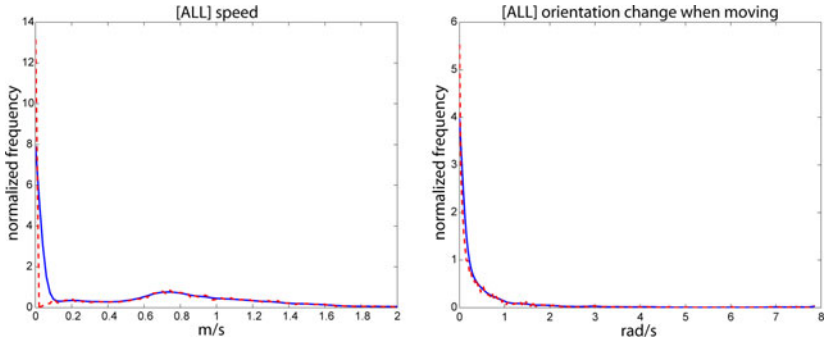
## 4.1 Dataset

The data used to extract the statistics has been kindly provided by Lerner *et al* [5]. The employed sequence shows a busy square from a stationary camera, oblique view, with a total of 450 subjects in 5400 frames. Most of the subjects walk from one of the borders of the scene to another and stay within the scene for about 15 seconds, while some stand longer in the scene talking to other subjects or waiting. An example frame is shown in Fig. 2. The sequence is particularly challenging due to low image resolution, interlacing and compression artifacts, cast shadows, as well as the large number of people. We manually annotated the head position of each subject and estimated a homography matrix to retrieve metric properties. In a second step, we annotated groups in the sequence, by relying on several cues, such as people talking to each other or holding hands, for example. For our purposes, we split the sequence in a training (3400 frames) and testing section (2000 frames).

## 4.2 Independent Motion and Appearance

Pedestrians change the walking direction smoothly. Furthermore, the walking speed is not arbitrary. This information is commonly exploited in motion prior

<sup>2</sup> Note that the estimation of the homography matrix introduces a small scalar factor compared to the real walking speed.



**Fig. 3.** Statistics over a person's movement: **Left:** the distribution  $P(s_i^t)$  over speeds shows two peaks for people standing and walking. **Right:** the figure shows  $P_{s_i^t \geq 0.15}(\alpha_i^t | \alpha_i^{t-1})$ . For walking people, there is a preference to keep the current heading. Red indicates the original data points, blue the histogram estimate.

for pedestrians in a constant velocity model. To model these factors we define the motion term of Eq. 3 as

$$\phi_i^{motion}(\mathbf{h}_i) = \sum_{t=0}^{T-1} \log[P_{s_i^t < 0.15}(\alpha_i^t | \alpha_i^{t-1}) + P_{s_i^t \geq 0.15}(\alpha_i^t | \alpha_i^{t-1})] + \sum_{t=0}^{T-1} \log P(s_i^t). \quad (4)$$

$P_{s_i^t < 0.15}(\alpha_i^t | \alpha_i^{t-1})$  is assumed uniform while  $P(s_i^t)$  and  $P_{s_i^t \geq 0.15}(\alpha_i^t | \alpha_i^{t-1})$  are estimated by building a normalized histogram (smoothed with a Gaussian kernel) of the angles and speeds extracted from the training set and are shown in Fig. 3. As one can expect, from the speed statistics it is easy to distinguish two modes, corresponding to standing and walking people. Fig. 3 shows also that the choice of  $0.15 \text{ m/s}$  for telling apart walking and standing pedestrian is a reasonable one.

For the appearance term, we directly use the output of the tracker (see Sec 6).

$$\phi_i^{app}(\mathbf{h}_i | \mathbf{I}) = \log f^{app}(\mathbf{h}_i | \mathbf{I}). \quad (5)$$

### 4.3 Grouping Relations

Given two pedestrians, one of the obvious features that makes it possible to guess whether they belong to the same group or not, is proximity. So, when two pedestrians belong to the same group, their distance is kept to a certain value. If they are walking, the estimate of the orientation can give us further information on how they are positioned with respect to each other. For two pedestrians belonging to the same group, we therefore define

$$\psi_{ij|l_{c(ij)}}^{pos}(\mathbf{h}_i, \mathbf{h}_j, l_{c(ij)} = 1) = \sum_{t=0}^{T-1} \log[P_{s_i^t \geq 0.15 \wedge s_j^t \geq 0.15}(\mathbf{p}_i^t | \mathbf{p}_j^t \alpha_j^t, l_{c(ij)} = 1, ) + P_{s_i^t < 0.15 \vee s_j^t < 0.15}(d(\mathbf{p}_i^t, \mathbf{p}_j^t) | l_{c(ij)} = 1)], \quad (6)$$

where  $d(\mathbf{p}_i^t, \mathbf{p}_j^t)$  is the Euclidean distance between the positions  $\mathbf{p}_i^t$  and  $\mathbf{p}_j^t$ .  $P_{s_i^t \geq 0.15 \wedge s_j^t \geq 0.15}(\mathbf{p}_i^t | \mathbf{p}_j^t, \alpha_j^t, l_{c(ij)} = 1)$  and  $P_{s_i^t < 0.15 \vee s_j^t < 0.15}(d(\mathbf{p}_i^t, \mathbf{p}_j^t) | l_{c(ij)} = 1)$  are estimated using histograms as before and are shown in Fig. 4. For pedestrians do not belong to the same group, we found it unnecessary to distinguish between walking or standing. The main feature, when dealing with the position of two individual pedestrians, seems to be the *repulsion* effect: individuals try not to come close to each other unless necessary. In this case, we define the motion term as

$$\psi_{ij|l_{c(ij)}}^{pos}(\mathbf{h}_i, \mathbf{h}_j, l_{c(ij)} = 0) = \sum_{t=0}^{T-1} \log P(d(\mathbf{p}_i^t, \mathbf{p}_j^t) | l_{c(ij)} = 0), \quad (7)$$

where  $P(d(\mathbf{p}_i^t, \mathbf{p}_j^t) | l_{c(ij)} = 0)$  is again estimated using histograms and shown in Fig. 4.

Another important feature of people when walking in the same group, is that they have the same orientation. We therefore define

$$\psi_{ij|l_{c(ij)}}^{ang}(\mathbf{h}_i, \mathbf{h}_j | l_{c(ij)} = 1) = \sum_{t=0}^{T-1} \log P_{(s_i^t \geq 0.15 \wedge s_j^t \geq 0.15)}(\alpha_i^t, \alpha_j^t | l_{c(ij)} = 1). \quad (8)$$

As before, this term is estimated with a smoothed histogram approach. The density is shown in Fig. 4 and, as expected, shows that subjects that walk together keep the same orientation. We did not observe an interesting orientation pattern among pedestrians that are not in the same group, therefore we assume uniform  $P_{(s_i^t \geq 0.15 \wedge s_j^t \geq 0.15)}(\alpha_i^t, \alpha_j^t | l_{c(ij)} = 0)$ .

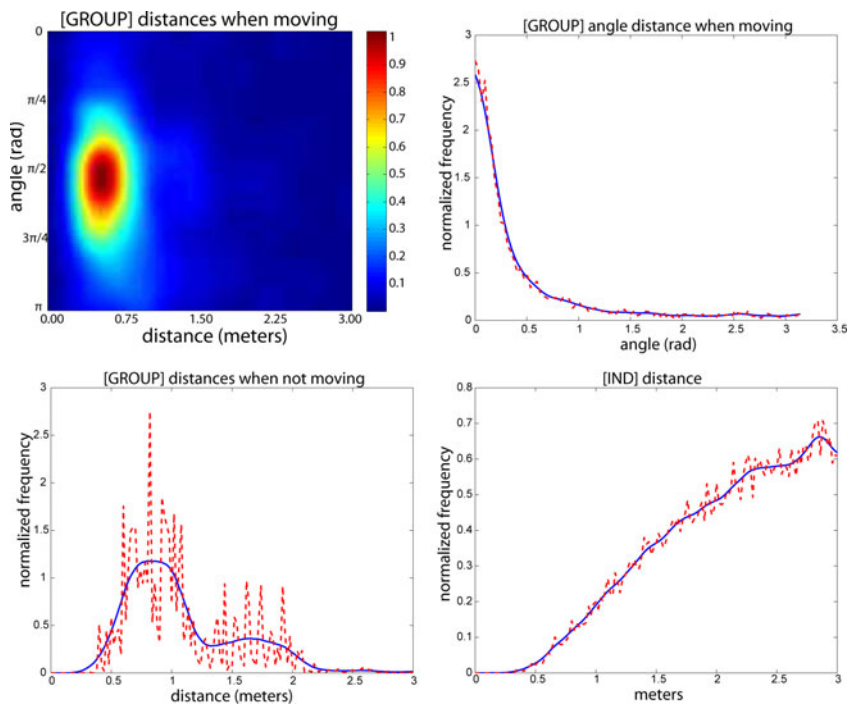
Finally,  $\gamma_{c(ij)}(l_{c(ij)})$  could be set by looking at the fraction of grouping relations over the total number of relations. Although the correct value for the fraction would be  $\sim 23\%$  for our dataset, we will vary this value to measure the robustness of our model (see Sec. 6).

#### 4.4 Transitivity Constraints

The hard constraint in Eq. 2 is modeled by penalizing impossible configurations with an opportunely large constant cost.

## 5 Inference

We are looking for the most probable joint assignment of the trajectories  $\mathbf{H}$  together with the grouping relations  $\mathbf{L}$  in Eq. 3. Exact inference is intractable, as the graph contains cycles and the potentials are not restricted to a particular kind (e.g., submodular). For the inference, we use Dual Decomposition (DD) [23], building on the code made available by [24]. DD optimizes the Lagrangian dual of the LP-relaxation of the original problem, by decomposing the problem into a set of subproblems, each of which can be solved efficiently. By optimizing the dual, it gives a lower bound that can be used to check whether the method converged to a global optimum (i.e. when the solution given by the primal has the same energy as the solution of the dual problem).



**Fig. 4.** Statistics over interacting people. **Top-left:**  $P_{s_i^t \geq 0.15 \wedge s_j^t \geq 0.15}(\mathbf{p}_i^t | \mathbf{p}_j^t, \alpha_j^t, l_{c(ij)} = 1)$  in polar coordinates, such that radius is the distance  $d(\mathbf{p}_i^t, \mathbf{p}_j^t)$  and the angle is the angle under which  $j$ , with absolute orientation  $\alpha_j^t$  sees  $i$ . When moving in groups, people keep a low distance from each other, trying to walk side by side. **Top-right** shows  $P_{(s_i^t \geq 0.15 \wedge s_j^t \geq 0.15)}(\alpha_i^t, \alpha_j^t | l_{c(ij)} = 1)$ . As expected, people that walk together are headed in the same direction. **Bottom-Left** shows  $P_{s_i^t < 0.15 \vee s_j^t < 0.15}(d(\mathbf{p}_i^t, \mathbf{p}_j^t) | l_{c(ij)} = 1)$ . The distribution is less peaked than the distribution shown in the top-left figure, probably reflecting the fact that when people are standing in groups, they allow for more flexible configurations. **Bottom-Right:** the figure shows  $P(d(\mathbf{p}_i^t, \mathbf{p}_j^t) | l_{c(ij)} = 0)$ . Like for groups, the repulsion effect between individuals, used in many pedestrian motion models [2, 11], is evident from the low value around 0.

In our case, we decompose the original graph first into a constraint layer containing only transitivity constraints factors and a data layer containing all the other factors. Then these sub-graphs are further decomposed into spanning trees. We optimize each tree separately using standard Belief Propagation [25]. The primal solution, and therefore the upper bound to the optimal solution, is found by using a heuristic similar to that described in [23].

## 6 Experiments

The proposed model requires a set of hypotheses to choose from. In this section, we therefore first describe how to build up the model given an input frame, before presenting experiments on real-world data.



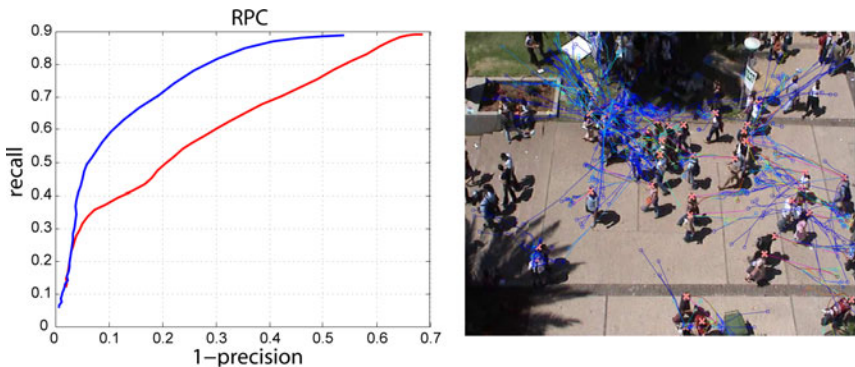
## 6.1 Model Construction

*Hypothesis Generation.* Given a starting frame  $t_0$ , a separate set of hypotheses  $\mathcal{H}_i$  is generated for each currently tracked person  $i$ . Each hypothesis  $\mathbf{h}_i$  describes a possible motion of the subject between  $t = t_0 \dots t_{T-1}$ . To this end, we start a single-person tracker for each person  $i$  at  $t_0$ , at each time step following the cost function recursively according to a best-first paradigm. Following at each time step  $t$  the  $M$  best options therefore yield a maximum of  $M^T$  hypotheses per person. As a cost function, we employ several cues: as a motion and appearance model, we use a constant velocity assumption, respectively an HSV-color histogram  $\mathbf{a}_i^t$  on the subject’s head. The product of the Bhattacharyya coefficients  $d(\cdot, \cdot)$  along the trajectory is then used to define  $f^{app}(\mathbf{h}_i | \mathbf{I}) = \prod_{t=1}^{T-1} d(\mathbf{a}_i^t | \mathbf{a}_i^{t-1})$ .

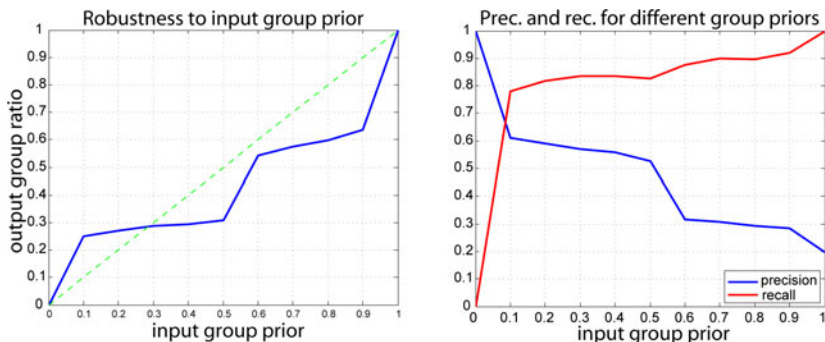
As a third cue, we consider a discrete set of detections in the current hypothesis’ vicinity. The detections are obtained from a voting-based detector [26], trained on both head and upper bodies from a total of 1145 positive and 1208 negative examples. Even though specifically trained on the same setup’s data, the detector only reaches an equal error rate of 0.65 (head) respectively 0.76 (torso) (see Fig. 5). The reason for this low performance is a higher number of false positives on strong cast shadows, as well as some false negatives when people are standing very closely together. To account for frequent false negatives, up to 50% of missing detections are allowed inside a trajectory, where the missing parts are interpolated using the constant velocity model.

To handle the case of persons leaving the scene, we introduce a set of virtual detections at the border of the image. Once a tracker selects such a detection, it is terminated, and the corresponding trajectory corresponds to a linear extrapolation starting from that time step.

For computational reasons, in the presented experiments, we use a time step of 0.2 seconds, and set  $T = 10$  (thus always considering time windows of 2 seconds) and  $M = 4$ , yielding an average of 147 hypotheses per subject. We run



**Fig. 5.** **Left:** RPC curves for head detector (red) and torso detector (blue). **Right:** Sample hypotheses for one frame, with blue corresponding to low confidence and red to high. Especially in crowded areas, many possible hypotheses can be generated.



**Fig. 6. Left:** Estimate of groups relations while varying the grouping prior  $\gamma$ . **Right:** precision and recall curves for group relations for different  $\gamma$  values.

the experiment each 2s for all pedestrians, starting from 40 different frames. This results in 1236 subjects being tracked.

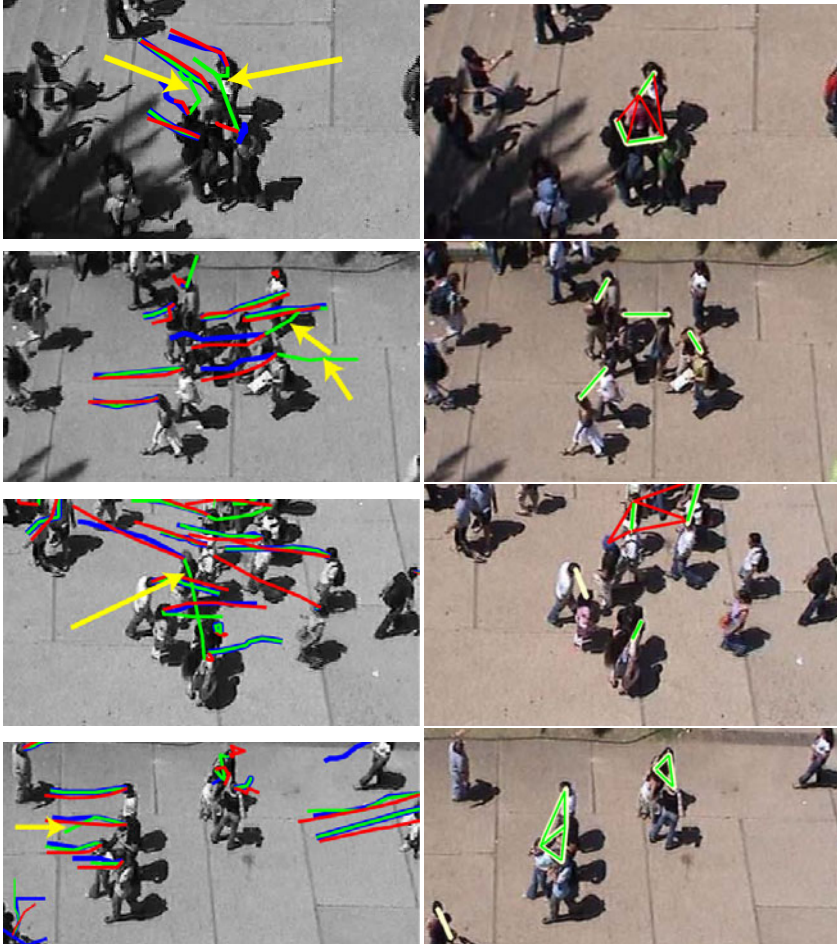
*Link Modeling.* To set up the links between individuals, Delaunay triangulation is performed on the subjects positions in the input frame. Links longer than 3 meters are cancelled.

## 6.2 Groundtruth

Before using an actual detector to drive hypothesis generation, we perform a baseline experiment, where we use the ground-truth annotations as detections (note that we are operating on the test sequence, i.e., the training of the model did not use this data at all). To measure the effect of the proposed model, we compare the output of the inference stage with locally selecting the best trajectories (i.e., the hypothesis with the maximum unary term). We report the number of correctly selected trajectories as the ones that coincide with the ground-truth completely. In Fig. 6 (left), we run this experiment for different values of the grouping prior  $\gamma$ . As can be seen, the model does not blindly trust the prior (unless set to the extreme positions), but moves towards the true fraction of groupings (0.23) disregarding the starting position. The performance of the model with respect to trajectory selection is hardly affected by the grouping: the unary makes 34 mistakes, whereas the full model, depending on the chosen grouping prior, performs considerably better with  $12 \pm 2$  mistakes. Only when  $\gamma = 0$ , our model makes 22 mistakes. In Fig. 6 (right), we furthermore plot recall and precision of finding groups, again varying over the prior  $\gamma$ . The numbers stay quite constant for a large range of  $\gamma$ , underlining the stability of the model. Choosing extreme values will naturally also lead to inferior results, either in favor of groups or not. In the upcoming experiments, we will use an uninformed prior,  $\gamma = 0.5$ .

**Table 1.** Performance of model when using raw detections as input. The proposed model not only improves the correctly chosen trajectories, but also recovers groups with high recall and good precision.

	Wrong Trajectories	Groups			
		TP	TN	FP	FN
Local	401	-	-	-	-
Group CRF	363	389	1526	449	84



**Fig. 7.** Example situations (close-ups). **Left:** trajectories, with ground truth (red) and solutions found by the unary term alone (green) and the group CRF (blue). **Right:** grouping, with ground truth (white), true/false positives (green/red) (see text).

### 6.3 Detector

When starting from a ground-truth point and generating hypotheses using detections, the generation step has to deal with a considerable number of false positives (generating excess wrong trajectories) and false negatives (in the worst case, missing an entire trajectory). Due to these inaccuracies, we change the notion of correct trajectory to an error  $< 0.5$  m from the ground-truth at the last trajectory position. The subject errors and the group statistics are reported in Table 1. Note that this experiment is considerably harder, so the number of errors in absolute terms increases. Still, our method improves  $\approx 10\%$  w.r.t. using only the unary terms, i.e. without grouping. The group statistics show a precision of 46% (about twice above the chance level of 23%) and 82% recall.

Some example images, comparing the two methods, are shown in Fig. 7. For each sample, we report both the trajectories found by either choosing the local optimum or the group CRF, as well as the recovered grouping by our model. In the top row, the grouping information gives a twofold improvement, encouraging the two persons to move together to the left side, as opposed to choosing intersecting trajectories (yellow arrows). One single wrong link between the two correctly inferred groups spurs the creation of additional wrong links through transitivity. In the second row, grouping correctly enforces the two people in the middle to walk together to the left as opposed to the local solution, which erroneously goes to the right (yellow arrows). In the third row, the joint reasoning keeps the group CRF from choosing the wrong path leading through all the pedestrians (yellow arrows), thus highlighting the spatial exclusion constraint. Finally, in the last row, grouping encourages smoother trajectories that stay well separated, with the group on the left correctly estimated.

## 7 Conclusions

In this paper we investigated the influence of pedestrian interactions on data association in crowded scenes, having in mind a tracking application. Statistics learned on natural video data show that people walking in groups behave differently from people walking alone. Commonly hard-coded effects such as repulsion/avoidance were also clearly visible in the data. These statistics were used to train a graphical model encoding the interactions between pedestrians in a principled manner. The model was optimized for the MAP estimate with a state of the art approximate inference engine, giving a joint estimate about correct trajectories and group memberships in the data.

The results show that interactions should be taken into account when reasoning about people trajectories. We not only showed that joint optimization is beneficial in terms of tracking error, but we were able to recover, with a good recall and a sufficient precision, group statistics.

The running time depends on the number of people in the scene. Our current implementation of the system, far from being optimized, takes few minutes ( $\approx 10$ ) to output trajectories of length 2 seconds and grouping relations.

The focus of this paper was rather the effect of interactions as opposed to a complete tracking application. We therefore only showed results on short time windows initialized from ground-truth locations, not forming entire trajectories automatically. Extending the model in this direction will be part of future work.

## References

1. Hall, E.T.: *The Hidden Dimension*. Garden City (1966)
2. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Physical Review* 51(5) (1995)
3. Penn, A., Turner, A.: Space syntax based agent simulation. In: *Pedestrian and Evacuation dynamics* (2002)
4. Schadschneider, A.: Cellular automaton approach to pedestrian dynamics - theory. In: *Pedestrian and Evacuation Dynamics* (2001)
5. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: *EUROGRAPHICS* (2007)
6. *Massive Software: Massive* (2010)
7. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 1–14. Springer, Heidelberg (2008)
8. Antonini, G., Martinez, S.V., Bierlaire, M., Thiran, J.: Behavioral priors for detection and tracking of pedestrians in video sequences. *IJCV* 69, 159–180 (2006)
9. Choi, W., Shahid, K., Savarese, S.: What are they doing? collective activity classification using spatio-temporal relationship among people. In: *Workshop on Visual Surveillance (VSWS '09) in conjunction with ICCV'09* (2009)
10. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using Social Force model. In: *CVPR* (2009)
11. Pellegrini, S., Ess, A., Schindler, K., Gool, L.V.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: *ICCV* (2009)
12. Scovanner, P., Tappen, M.: Learning pedestrian dynamics from the real world. In: *ICCV* (2009)
13. Ge, W., Collins, R., Ruback, B.: Automatically detecting the small group structure of a crowd. In: *IEEE Workshop on Applications of Computer Vision, WACV* (2009)
14. French, A.: *Visual Tracking: From An Individual To Groups of Animals*. PhD thesis (2006)
15. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: *ICCV* (2009)
16. Ess, A., Leibe, B., Schindler, K., Gool, L.V.: A mobile vision system for robust multi-person tracking. In: *CVPR* (2008)
17. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: *CVPR* (2009)
18. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
19. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. *IJCV* 75, 247–266 (2007)
20. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: *CVPR* (2008)

21. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI* 27(11), 1805–1819 (2005)
22. Sutton, C., McCallum, A.: Piecewise training of undirected models. In: *Conference on Uncertainty in Artificial Intelligence, UAI (2005)*
23. Komodakis, N., Paragios, N., Tziritas, G.: MRF optimization via dual decomposition: Message-passing revisited. In: *ICCV (2007)*
24. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II. LNCS*, vol. 5303, pp. 596–609. Springer, Heidelberg (2008)
25. Mooij, J.M., et al.: libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference (2010), <http://www.libdai.org/>
26. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: *CVPR (2009)*

# Globally Optimal Multi-target Tracking on a Hexagonal Lattice

Anton Andriyenko<sup>1</sup> and Konrad Schindler<sup>1,2</sup>

<sup>1</sup> Computer Science Department, TU Darmstadt

<sup>2</sup> Photogrammetry and Remote Sensing Group, ETH Zürich

**Abstract.** We propose a global optimisation approach to multi-target tracking. The method extends recent work which casts tracking as an integer linear program, by discretising the space of target locations. Our main contribution is to show how dynamic models can be integrated in such an approach. The dynamic model, which encodes prior expectations about object motion, has been an important component of tracking systems for a long time, but has recently been dropped to achieve globally optimisable objective functions. We re-introduce it by formulating the optimisation problem such that deviations from the prior can be measured independently for each variable. Furthermore, we propose to sample the location space on a hexagonal lattice to achieve smoother, more accurate trajectories in spite of the discrete setting. Finally, we argue that non-maxima suppression in the measured evidence should be performed during tracking, when the temporal context and the motion prior are available, rather than as a preprocessing step on a per-frame basis. Experiments on five different recent benchmark sequences demonstrate the validity of our approach.

## 1 Introduction

Multi-target tracking in video sequences is a fundamental task of computer vision and video processing, with applications in surveillance, semantic video search, driver assistance, and many more. From a high-level point of view, the aim is to estimate the spatial trajectories of a number of targets over time, i.e. the task is solved when the locations of all targets at each time step are known.

Compared to single-target tracking, the multi-target problem poses additional difficulties: *data association* needs to be solved, i.e. it has to be decided which observation corresponds to which target; and *constraints* between targets need to be taken into account – most importantly, no two targets can occupy the same space at the same time. In probabilistic terms, one aims to maximise the joint posterior of several variables, which are not independent. That posterior depends on two factors: an *observation model*, which measures the agreement between the observed image evidence and the expected appearance of a target; and a *dynamic model*, which measures the agreement between a trajectory and the expected motion pattern of a target.

In the recent past, a main research challenge in multi-target tracking has been to develop schemes which are able to find (nearly) *global* maxima of the

posterior over the set of trajectories. Their common characteristic is that, in order to enable global optimisation, the set of permissible target locations has to be restricted to a manageable finite set. A-priori the set of possible locations is infinite, or at least very large. There are two main strategies to restrict it to a reasonably small set: either candidate locations are found by thresholding and/or non-maxima suppression of the observation likelihood; or the tracking region is sampled on a regular grid.

The dominant strategy so far has been the first one: the image evidence – typically the output of object detection or background subtraction – is used to identify the most promising target locations per frame. These serve as input for the tracker, which links them to trajectories. A limitation of this strategy is that candidate locations are implicitly assumed to correspond perfectly with true target positions; there is no concept of localisation uncertainty. Another problem is that the space is sampled *only* at promising locations, hence target locations are not even defined in case of missing evidence (e.g. if two targets were both missed by the observation model, it is no longer checked whether they would collide in that frame).

The regular discretisation is attractive, because it is more generic, and because it avoids intermediate hard decisions based on partial evidence, thus allowing for principled probabilistic modelling. A disadvantage is that to keep tracking computationally tractable the grid needs to be significantly coarser than typical image resolutions, and therefore introduces aliasing. A particularly undesirable consequence of the discretisation is that the space is no longer isotropic – the smoothness of a trajectory depends on its alignment with the grid, and jagged trajectories complicate the usage of reasonable dynamic models, which favour smooth motion.

In this paper, we present a global optimisation approach to multi-target tracking on a regular grid, with an a-priori *unknown* number of targets. Original contributions of the work are

- We “re-introduce” the dynamic model, which has traditionally been an integral part of tracking, but in previous work had to be dropped to achieve objective functions, which can be solved to (near) global optimality. Specifically, we include the constant heading prior.
- To best utilise the dynamic model and achieve smoother, more accurate trajectories despite the discrete setting, we propose to use a hexagonal sampling of the location space, rather than a rectangular one.
- We perform non-maxima suppression during tracking rather than independently in every frame, allowing the tracker to recover the most likely locations in the light of *all* evidence, rather than the locally best guess per frame.

Despite the proposed extensions the resulting maximisation of the posterior can still be written as an integer linear program (ILP), by an extension to the formulation of [1]. The ILP is solved efficiently through a linear programming relaxation, in most cases to global optimality.



## 2 Related Work

Multi-target tracking algorithms can be roughly classified as either recursive methods which base their estimate only on the state of the previous frame, or methods which seek optimality over an extended period of time. Recursive methods rely on a first-order Markov assumption, usually using either Kalman filtering, e.g. [23], or – in the presence of more complex posteriors – particle filtering, e.g. [456]. A different strategy is to aim for an optimal solution over multiple frames. To this end the state space is restricted to a discrete number of possible target locations, either by heuristics based on the single-frame target likelihood, e.g. [78910], or by sampling locations on a regular grid, e.g. [111].

The more popular strategy has so far been to use single-frame heuristics. After measuring the likelihood that a target is present at any given image location – in most cases by variants of object detection [1213] or background subtraction [14] – the likelihood function is thresholded and/or its local maxima are found; possible object locations are restricted to these maxima. The optimisation then chooses the best set of trajectories over time, based on the selected locations. Depending on the formulation, this leads to an ILP which is solved by relaxation [8], an integer quadratic program which is solved with problem-specific search procedures [7], or a network flow problem [10], which is in fact closely related to ILP, c.f. [15]. The pruning strategy would be entirely sufficient if the per-frame processing were entirely correct. In practice it has an important shortcoming: the evidence will never be perfect, so that the discrete set after pruning will suffer from false positives (spurious maxima), false negatives (missing maxima), and localisation errors (displaced maxima).

To still restrict the state space, without relying on the per-frame measurements, it has therefore been proposed to sample locations on a regular grid rather than at the modes. Research into grid-based trajectory optimisation started with methods which greedily aim for an optimal trajectory *per target*, e.g. using dynamic programming [11]. In the radar tracking literature, it has been shown long ago how to extend the dynamic programming approach to simultaneously track multiple targets [16], however in practice the computational complexity is prohibitive. An important step forward, which has also inspired our work, is the recent work of Berclaz et al. [1]. Tracking is performed in a globally optimal manner on a regular grid, by casting the problem as an ILP, again solved through relaxation. Contrary to [8] the number of targets need not be known in advance, which is achieved by adding source and sink nodes that can spawn, respectively terminate, trajectories, similar to [10].

This elegant formulation has two main limitations: firstly, in order to arrive at the ILP, the dynamic model had to be discarded. Object dynamics, which are an important component of tracking, are included only in a simplistic way, by allowing arbitrary motion within a grid point's 8-neighbourhood. Secondly, we found that a very peaky observation likelihood with sharp maxima at single grid locations is required for the method to work well. As soon as the object evidence

is blurry and “connecting the dots” becomes ambiguous, the method tends to instantiate multiple trajectories for the same target. This is a price the method pays for the desirable property that it allows for a variable number of targets.

The goal of the present work is to remedy these shortcomings by extending the model appropriately, while still keeping it linear, and hence amenable to global optimisation.

### 3 Model

In the following we give a detailed description of the proposed multi-view tracking method. We start with the formulation of maximum a-posteriori trajectory estimation as an integer linear program, which is an extension of the formalism introduced by [1]. Next, we introduce our observation model, a probabilistic variant of *tracking-by-detection* designed for tracking targets observed from multiple viewpoints in world coordinates. Furthermore we propose to include non-maxima suppression in the tracker, rather than viewing it as a preprocessing step. We then write the dynamic model as a local soft constraint, by penalising the changes between consecutive motion vectors. In this form it can be re-introduced into the ILP-formulation of multi-target tracking. Finally we move to an important technical issue: in the discrete setting the dynamic model suffers from grid aliasing, hence it is a lot more effective to quantise locations to a hexagonal rather than a rectilinear grid.

#### 3.1 Tracking as Integer Linear Program

To set the stage, we extend the ILP-formulation of multi-target tracking introduced in recent work [8,10,11] for our purposes. The possible target locations are discretised to a finite set of sites  $\mathbf{x}_i = (x_i, y_i)$ . Among those sites a neighbourhood system  $\mathcal{S}$  is defined, where a site’s neighbours  $\{\mathbf{x}_j : j \in \mathcal{S}(i)\}$  are all sites that can be reached from  $\mathbf{x}_i$  in a single time step, including  $\mathbf{x}_i$  itself.

Next, we define a *tracklet*  $X_{ijk}^t$  as an allowable path over 3 consecutive frames, i.e. a set of three sites  $X_{ijk}^t = \{\mathbf{x}_i^{t-1}, \mathbf{x}_j^t, \mathbf{x}_k^{t+1}\}$  such that  $j \in \mathcal{S}(i)$  and  $k \in \mathcal{S}(j)$ . The set of all index triplets  $(ijk)$  that produce a valid tracklet is denoted  $\mathcal{T}$ . The tracklets are the variables of our optimisation problem. They take on values  $X_{ijk}^t \in \{0, 1\}$ , where  $X_{ijk}^t = 1$  means that tracklet  $(ijk)$  is part of some trajectory, and  $X_{ijk}^t = 0$  means that it is not part of any. The reason for introducing the tracklets is that the dynamic model cannot be included efficiently when operating directly on the sites  $\mathbf{x}_i^t$ , as will become clear in Sec. 3.4.

Based on the observed evidence  $\mathbf{R}$ , each tracklet is assigned a goodness-of-fit  $u_{ijk}^t = \log \frac{P(X_{ijk}^t=1|\mathbf{R})}{P(X_{ijk}^t=0|\mathbf{R})}$  which compares the hypotheses  $X_{ijk}^t = 1$  and  $X_{ijk}^t = 0$  in the light of the observation model (Sec. 3.2) and the dynamic model (Sec. 3.4). Thus, multi-target tracking becomes maximising the posterior by picking the best set of tracklets  $\mathbf{X}^*$  from  $\mathcal{T}$ , under two constraints:

1. *collision avoidance*: no two tracklets can have the same midpoint  $\mathbf{x}_j^t$ ; whenever a tracklet  $X_{ijk}^t$  is selected, all other tracklets  $X_{qjr}^t$  must be discarded.

2. *continuity*: tracklets must form continuous trajectories – whenever a certain tracklet is used in a solution,  $X_{ijk}^t = 1$ , there must be exactly one tracklet  $X_{jkl}^{t+1}$  in the next time step, which is also used. Targets entering or leaving the tracking area are modelled by two virtual *source* and *sink* sites, which are neighbours of all boundary sites and can emit, respectively absorb, targets.

The MAP estimation amounts to the following optimisation problem with the vector  $\mathbf{X}$  of all tracklets  $X_{ijk}^t$  as argument:

$$U^* = \max_{\mathbf{X}} \sum_{ijk \in \mathcal{T}, t} (u_{ijk}^t \cdot X_{ijk}^t) \tag{1}$$

$$\text{s. t. } \forall ijk \in \mathcal{T}, t : \sum_{q:qjk \in \mathcal{T}} X_{qjk}^t = \sum_{r:jkr \in \mathcal{T}} X_{jkr}^{t+1} \quad (\text{continuity}) \tag{2}$$

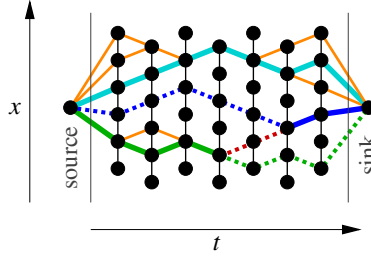
$$\sum_{q,r:qjr \in \mathcal{T}} X_{qjr}^t \leq 1 \quad (\text{collision avoidance}) \tag{3}$$

$$X_{ijk}^t \in \{0, 1\} \quad (\text{domain of variables}) \tag{4}$$

*Optimisation.* Maximising Eq. (1) is an integer linear program, and hence NP-complete. However, it can be relaxed to a linear program by replacing the condition  $X_{ijk}^t \in \{0, 1\}$  with  $0 \leq X_{ijk}^t \leq 1$ . The relaxed problem can be efficiently solved with the simplex algorithm or an interior-point method. Moreover, if all variables  $X_{ijk}^t$  at the relaxed optimum  $\mathbf{X}_{LP}^*$  take on integer values, then it is also a global optimum of the original problem,  $\mathbf{X}_{LP}^* = \mathbf{X}_{ILP}^*$ . In practice, this happens in most cases. Even if the solution is not completely integral, then in practice the optimality gap is small, and only a tiny fraction of non-integer variables remains (in our experiments  $< 0.2\%$ ), and these are clustered in relatively small connected components of the neighbourhood system. Hence, an optimum of the ILP can be found using the branch-and-cut method with the relaxation as bounding function (“mixed integer programming”), or by “probing”, i.e. rounding some non-integer values and solving for the others while monitoring the objective value  $U$  (a similar strategy is known as QPBO-P in the graph-cuts context [7]).

To gain some intuition why the LP-solution  $\mathbf{X}_{LP}^*$  is largely integral, and amenable to probing or bounding, it is instructive to look at the behaviour of simple paths connecting the source to the sink (see also Fig. 1):

- trivially, all tracklets on a junction-free path  $Q$  have the same value  $X_Q$  because of the continuity constraint;  $X_Q$  will always be integral, because the total contribution of the path to the objective value is  $X_Q \sum u_Q$ , which attains its maximum at  $X_Q=0$  for  $\sum u_Q \leq 0$ , and at  $X_Q=1$  for  $\sum u_Q > 0$ .
- if a path were to split into two branches  $Q$  and  $R$  at any point (including the source) and converge again at a later point (including the sink), then one branch would get all the weight, whereas the other would be suppressed: the total contribution of the two branches is  $X_Q \sum u_Q + (1 - X_Q) \sum u_R$ , which attains its maximum at either  $X_Q=1$  or  $X_Q=0$ .



**Fig. 1.** LP-relaxation of multi-target tracking. On a junction-free path from source to sink (cyan), all variables are  $X_{...} = 1$ . Branching *within* a path is impossible (e.g. orange paths must have  $X_{...} = 0$ ). A bridge (red) which permits to shift weight from one path (green) to another (blue) may cause non-integer values in the dashed regions.

- the branching argument applies recursively, so non-integer values can only occur when *two different paths* are connected by a “bridge”, so that weight can be shifted from one to the other when their relative likelihood changes.

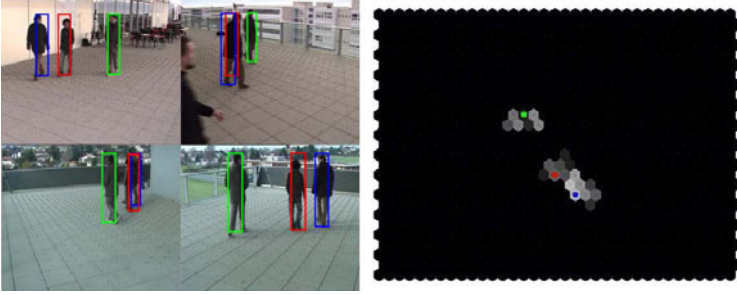
The solution  $\mathbf{X}_{ILP}^*$  of the ILP is the maximum a-posteriori set of trajectories over the observed time window  $\Phi$ . In practice this time interval is bounded by the available storage and computation power. The number  $M$  of variables and constraints to be stored grows linearly with  $\Phi$ , and the average-case computational complexity of LP-solvers is  $\mathcal{O}(M)$ , too. A practical solution is to solve Eq. (II) for overlapping time intervals and constrain the solutions to be consistent by fixing the first frame. Empirically, intervals of  $\Phi = 30$  frames are sufficient.

### 3.2 Observation Model

We formulate tracking in world coordinates for the general case of multiple cameras observing the scene from different viewpoints. Multi-camera setups greatly improve tracking accuracy when the camera positions are low over the ground, such that one has to accept inaccurate depth estimates as well as frequent occlusions. The framework includes single-view tracking as a special case, by setting the number of cameras to 1. As usual, the posterior is split into an observation likelihood and a motion prior. We further decompose the observation into two parts measuring object detection response, respectively colour similarity:

$$P(X_{ijk}^t = 1 | \mathbf{R}) \sim P_O(\mathbf{R} | X_{ijk}^t = 1) \cdot P_A(\mathbf{R} | X_{ijk}^t = 1) \cdot P(X_{ijk} = 1). \quad (5)$$

*Object detection.* To measure the support of targets in the image data, we use our own implementation of the popular HOG detector [13]. The detector scans the images  $I_\nu^t$  (taken from viewpoints  $\mathbf{c}_\nu$  at all three frames of the tracklet) over all positions  $\mathbf{u}$  and scales  $s$  with a binary classifier trained to discriminate people from background, and returns for every location and scale a classification score  $R_\nu^t$ . The scores are mapped from image locations  $(\mathbf{u}, s)$  to locations  $\mathbf{x}$  and target heights  $h$  in the world coordinate system with appropriate projections, and aggregated over all views and the three frames to obtain the total evidence  $\mathbf{R}$  for a tracklet.



**Fig. 2.** The evidence  $P(\mathbf{R}|X_{ijk}^t)$  has smooth peaks, which are not precisely localised. (left) tracking results in four views. (right) birds-eye view of the scene. Note that the correct position for the green subject is *not* the one with the highest score. Our algorithm avoids per-frame decisions and chooses the best location during tracking.

The evidence at this point depends not only on  $X_{ijk}^t$ , but also on the person height  $h$ , via the detection scale  $s$ . In principle one could track directly in the  $(\mathbf{x}, h)$ -space, with a constraint that the height of any given person should not change over time. To reduce the computational burden, we prefer to place a Gaussian prior  $P(h) = \mathcal{N}(h; \bar{h}, \sigma_h)$  on the person height and marginalise it out,

$$P_O(\mathbf{R}|X_{ijk}^t = 1) = \sum_q (P_O(\mathbf{R}|X_{ijk}^t = 1, h_q) \cdot P(h_q)). \quad (6)$$

*Appearance.* The generic object model is complemented with a target-specific appearance model to better distinguish different targets. To this end, we demand that the colour distribution of a target varies slowly over short time spans. All sites of a tracklet  $X_{ijk}^t$  are projected back to the respective image locations  $\mathbf{u}$ , and at each location a colour histogram is extracted. The histograms of consecutive sites in a tracklet are then compared with the Bhattacharyya distance  $d_B$ , and the results are combined over all pairs of sites and all viewpoints  $\mathbf{c}_\nu$  :

$$P_A(\mathbf{R}|X_{ijk}^t = 1) \sim \prod_{\mathbf{c}_\nu} \exp\left(-\frac{d_B(\mathbf{u}_i^{t-1}, \mathbf{u}_j^t) + d_B(\mathbf{u}_j^t, \mathbf{u}_k^{t+1})}{\sigma_B^2}\right) \quad (7)$$

### 3.3 Exclusion Constraints

Exclusion constraints between different tracklets ensure plausible interactions between the targets. The simplest form of constraint, which has been widely used in multi-target tracking, is the *collision avoidance* implemented by Eq. (3). We argue that exclusion constraints can also be applied over larger neighbourhoods, to incorporate non-maximum suppression (NMS) in the tracking framework rather than do it at the frame level, such that the retained location is the one which is optimal for the entire time interval, rather than for a single frame.

A main limitation of most tracking schemes is that non-maxima suppression is carried out on a per-frame basis. The evidence  $P(\mathbf{R}|X_{ijk}^t)$  measured by the observation model is in practice not a set of perfect spikes, but a smooth distribution

with peaks which are not well localised, see Fig. 2. To remedy this, the distribution is replaced by the modes only, found by some mode-seeking procedure like mean-shift or morphological erosion. Traditional non-maxima suppression thus commits to a location without taking into account the fact that target locations should be consistent over time. Instead, we propose to integrate NMS into tracking, rather than detection: the detector output is left to be ambiguous around the modes, and the optimisation can choose which location is most likely, given also evidence from neighbouring frames and the dynamic model. However, in this context an additional difficulty arises. Since the number of targets is not known a-priori, the evidence for a target at its neighboring locations can still be strong enough to generate multiple tracks. In other words, a prior is required, which formalises the intuition that plaits of intertwined trajectories are unlikely. To this end, we introduce a number of additional constraints, which prohibit not only collisions of targets at the *same* location, but also tracklets starting at immediately neighbouring locations (which amounts to the assumption that the grid sampling distance is smaller than the minimal possible distance between two targets),

$$\forall ijk \in \mathcal{T}, t : r \in \mathcal{S}(i) \Rightarrow X_{ijk}^t + X_{rjk}^t \leq 1 \quad (8)$$

These constraints prevent targets from moving too close to each other, and also avoid trajectories crossing in such a way that a collision would happen in the empty space between two grid locations.

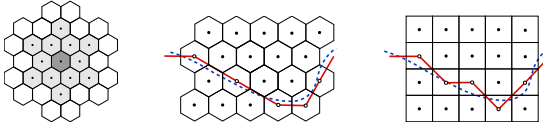
We point out that the effect of the prior is not the same as single-frame NMS: under the exclusion constraints the optimisation is free to choose a target location  $\mathbf{x}^t$ , which is *not* a maximum of the detection score in frame  $t$ , in order to achieve a smoother trajectory, or to avoid collisions with other targets.

### 3.4 Dynamic Model

An important ingredient of tracking is the dynamic model, which encodes prior knowledge about likely motion patterns of the tracked objects. Using such dynamic models – mostly assuming constant heading, constant velocity or constant acceleration – has a long and successful tradition, however such models have been dropped in grid-based tracking.

To overcome this, we extend the grid-based formulation to incorporate the *constant heading* model, i.e. we assume that objects tend not to change their motion *direction*. A prerequisite for the ILP formulation is that the objective function Eq. (1) be linear. To preserve the linearity, the motion prior  $P(X_{ijk}^t = 1)$  must be formulated such that it can be computed *locally for each variable* (i.e. its contribution must be part of the unary terms). This is the reason why we have introduced the *tracklets*: checking for constant heading requires two consecutive motion vectors, and hence three consecutive sites, thus the variables must cover at least three consecutive frames.

Given the two motion vectors  $\mathbf{m}_{ij} = (x_j - x_i, y_j - y_i, 1)^\top$  and  $\mathbf{m}_{jk} = (x_k - x_j, y_k - y_j, 1)^\top$  in a tracklet  $X_{ijk}^t$ , one can include the prior by penalising the heading change  $\alpha$  between them, measured in  $(x, y, t)$ -space. The tracklet is assigned a



**Fig. 3.** (left) The 12-neighbourhood and symmetry axes in a hexagonal tiling. (middle, right) Aliasing of an example trajectory on a hexagonal grid and a rectangular grid with the same sample density.

probability which grows inversely with  $\alpha^2(X_{ijk}^t)$ , such that deviations from the constant-heading assumption are penalised, as desired:

$$P(X_{ijk}^t = 1) \sim \exp\left(-\frac{\alpha^2}{\sigma_\alpha^2}\right) \quad \text{where} \quad \alpha = \arccos \frac{\mathbf{m}_{ij}^\top \mathbf{m}_{jk}}{|\mathbf{m}_{ij}| |\mathbf{m}_{jk}|} \quad (9)$$

Note that the angle  $\alpha$  is computed in  $(x, y, t)$ -space. The method can be trivially extended to favour constant *velocity* by penalising the difference between  $\mathbf{m}_{ij}$  and  $\mathbf{m}_{jk}$ , however we found the angle to work better, probably because of the varying step-length on a discrete grid.

The obvious effect of the dynamic model is that smoother, more accurate trajectories are estimated in the presence of inaccurate or weak evidence. Beyond its original purpose, the dynamic model also has a more subtle benefit on the optimisation: by penalising tracklets with strong heading changes, the motion prior sharpens the posterior, and thus the objective function  $U$ . As a consequence, the relaxation gap narrows, and fewer non-integer values occur. This effect is particularly strong in difficult circumstances, when the evidence  $P(\mathbf{R}|X_{ijk}^t = 1)$  is rather flat, such that the potential target locations spread out over a large number of tracklets. Therefore the dynamic model drastically reduces computation time (in our experiments by at least a factor of 10). In some cases the number of non-integer values without motion prior even becomes so high that it is no longer tractable to find an integral solution with branch-and-cut or probing.

### 3.5 Hexagonal Discretisation

To make tracking amenable to global optimisation with ILP, in the spirit of [8, 11], the location space  $\mathbf{x}$  must be discretised to a finite set of locations. As explained above, we prefer not to heuristically prune the per-frame likelihood  $P(X_{ijk}^t | \mathbf{R})$  to a small set of permissible locations, but rather sample the ground plane in a regular lattice. A natural choice, which has been used in previous work, is a rectilinear grid, similar to the image grid. Unfortunately, such a grid has a strong preference for the two canonical directions along the  $x$ - and  $y$ -axes, whereas target trajectories in other directions exhibit severe aliasing.

Aliasing is not a big problem in the absence of a dynamic model, but together with the proposed motion model it creates difficulties: to check the deviation from constant heading *locally* one needs to rely on the vectors between the grid locations, thereby penalising trajectories which are not grid-aligned and hence continuously change directions. To alleviate this effect and boost the positive

effect of the dynamic model, we propose to use instead a hexagonal tiling of the ground plane, inducing a tri-axial neighbourhood system. In this grid, the 8-neighbourhood is replaced by a 12-neighbourhood, which reduces staircasing artifacts, and allows one to better enforce the constant heading assumption, see Fig. 3. The hexagonal tiling has been used in other contexts in image processing and computer vision [18,19], precisely because it has more preferred directions and reduces aliasing artifacts. Note that the change of sampling grid does not impair data quality: the transformation is performed when mapping the target probabilities from images to the world coordinate system, so there is no additional resampling step that would further blur the data.

## 4 Experiments

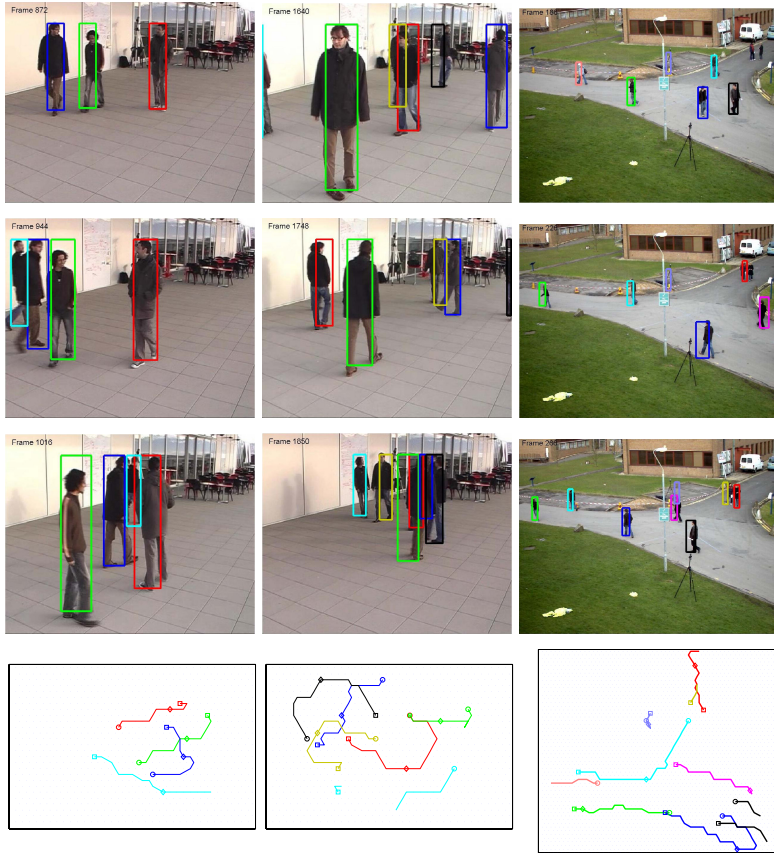
We present experiments on five different public multi-view video sequences. Sequences *campus-1* and *campus-2* [11] were both recorded from 3 different camera viewpoints, and have 2000, respectively 1400 frames showing up to 6 people moving outdoors. Sequences *terrace-1* and *terrace-2* [20] were both recorded from 4 viewpoints, and have 2000 frames each with up to 6 people, also moving freely outdoors. Finally, as a benchmark for monocular tracking we use sequence *PETS-S2L1* from the VS-PETS 2009 benchmark. The sequence is better suited for single-view tracking because of the elevated viewpoint. There are 795 frames showing up to 8 people moving in a street. The entire dataset contains 52 individual trajectories, which were manually annotated and used as ground truth. Due to the low target speed, we processed only every other frame of *PETS-S2L1* and every 6<sup>th</sup> frame in the remaining four sequences, such that targets move approximately one grid unit from one frame to the next.

All experiments have been carried out with the same set of parameters. The two free parameters of our method are the standard deviations  $\sigma_\alpha$  and  $\sigma_B$ , which govern the relative influence of detection score, colour similarity, and dynamic model (c.f. Sec. 3.2 and 3.4). To keep the optimisation tractable for long sequences, we follow the usual strategy and process overlapping time windows. This adds two further parameters, the number of frames  $\Phi$  per window, and the overlap  $\Omega$ . We set  $\Phi = 30$  (when processing every 6<sup>th</sup> frame at 25 fps, this amounts to  $\approx 7$  seconds) and  $\Omega = 10$ .

Figure 4 shows example results. Targets are tracked successfully over many frames, new targets entering the scene are initialised automatically. Especially the second example shows many targets moving in a small space. People are often occluded simultaneously in several views. Long-term occlusion is a main cause of failure, such as for the person marked in cyan.

In the *PETS-L2S1* sequence, up to 7 targets are tracked in *monocular* video over a large area of interest. Note the false positive on the tripod near the image centre: false detections on background objects are the dominant cause of false positives, since they tend to appear frequently on the same structures and, being static, fulfil the constraints of the dynamic model.

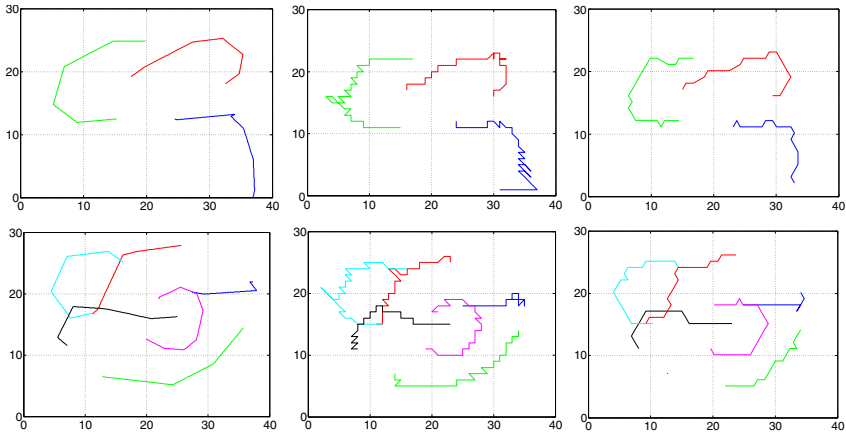




**Fig. 4.** Tracking results obtained with our algorithm. The left and middle column are from the *terrace-1* sequence, the right column is from *PETS-L2S1*. Displayed are three sample frames (1<sup>st</sup>-3<sup>rd</sup> row), and a birds-eye view of target trajectories (last row). The displayed frames are marked (top ○, middle ◇, bottom □). See text for details.

#### 4.1 Comparison to Previous Work

We directly compare the trajectories estimated by our method to those of [1], which we extracted from their published results. Their method is based on a similar ILP formulation, but on a rectilinear grid without dynamic model. Fig. 5 shows sample trajectories from both methods, with similar grid resolutions. The examples illustrate how late non-maxima suppression, together with the dynamic model, avoids implausible jittering. We emphasise that the improvement is due to the combination of all modelling choices: late non-maxima suppression preserves the necessary evidence for flexible target placement, while the dynamic model on a hexagonal grid supplies the constraints to handle the extra flexibility.

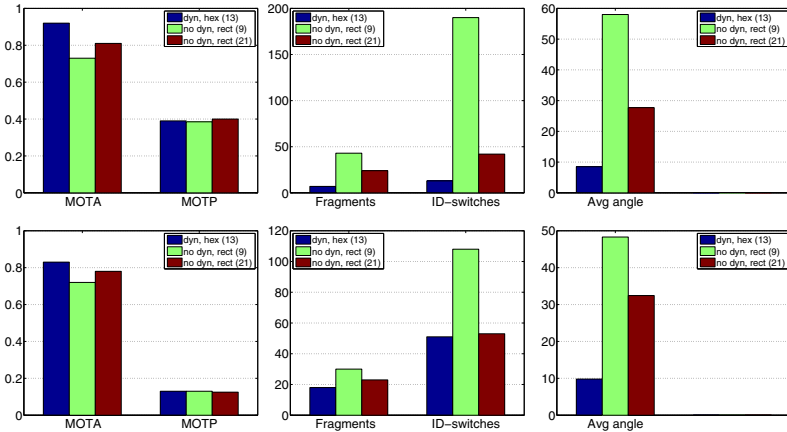


**Fig. 5.** Improved trajectories with the proposed model. (left) manually annotated ground truth for 200 frames of sequence *terrace-1*. (middle) trajectories reconstructed by state-of-the-art tracking *without* dynamic model [1]. (right) trajectories estimated by our system with dynamic model on a hexagonal grid.

## 4.2 Quantitative Evaluation

In the following we quantitatively evaluate our tracker against the baseline ILP tracker without dynamic model and operating on a rectilinear grid with either the standard 9-neighbourhood (8 neighbours and the central location itself) or a larger 21-neighbourhood. We use several metrics: on one hand we compute the CLEAR metrics for multi-object tracking [21] because of their growing popularity; on the other hand we count the number of trajectory fragments and ID switches, similar to [22]. Finally, we also measure the smoothness of the estimated trajectories by the average angle between the segments of all tracklets.

The evaluation results are summarised in Figure 6. As expected our model greatly improves trajectory smoothness, with a three- to five-fold reduction in the average tracklet angle for both multi-view and monocular tracking. The smoother trajectories also improve tracking accuracy: CLEAR-MOTA (measuring false negatives, false positives, and miss-matches) increases by 10-20%, because our model mitigates the effect of inaccurate and uncertain evidence. Using 21 instead of 9 neighbours also improves accuracy, but is still inferior to our result, while taking  $\approx 5$  times longer to compute due to the larger number of variables. Tracking precision (CLEAR-MOTP, measuring overlap of bounding boxes) improves insignificantly, because the metric is dominated by the alignment error due to the discrete location grid. At the same time, there is a dramatic reduction of fragmented tracks and identity switches ( $\approx 50\%$  for the monocular case, 80-90% for the multi-view case). Trajectory fragments are generated when the tracker drifts away from a target, which is less likely if late non-maxima suppression and the motion prior can correct inaccuracies of the evidence. ID switches happen when data association fails for targets very close to each other. The



**Fig. 6.** Tracking performance. (*left*) CLEAR metrics – higher is better. (*middle*) fragmentation and ID switches – lower is better. (*right*) smoothness – lower is better. Globally optimal tracking benefits significantly from dynamic models on the hexagonal grid, both in multi-view (top row) and in the monocular setting (bottom row).

motion prior improves correct data association, because it favours the option with more plausible dynamics.

## 5 Conclusion

We have presented an algorithm for tracking a varying number of targets on a discrete location grid. Multi-target tracking is cast as integer linear programming, and solved through LP-relaxation, in most cases to global optimality. Compared to previous research in this direction, we have argued that tracking should use the original target evidence as input and perform non-maxima suppression during trajectory estimation, and we have demonstrated how to include standard dynamic models in the ILP formulation. We have also shown that best results are achieved on a hexagonal rather than a rectilinear grid.

The experimental comparison on public benchmark videos confirms that beyond its theoretical appeal the proposed formulation delivers better tracking results and achieves superior performance in quantitative comparisons.

In future work we plan to analyse the cases, in which the relaxation alone does not return a global optimum. We believe that the problem structure can be exploited to solve those cases more efficiently. We also plan to use the result of the method as initialisation for a continuous optimisation scheme to overcome limitations due to the restriction to the discrete grid.

## References

1. Berclaz, J., Fleuret, F., Fua, P.: Multiple object tracking using flow linear programming. In: Winter-PETS (2009)

2. Black, J., Ellis, T., Rosin, P.: Multi-view image surveillance and tracking. In: Workshop on Motion and Video Computing (2002)
3. Mittal, A., Davis, L.: M<sup>2</sup>Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *Int. J. Comput. Vision* 51, 189–203 (2003)
4. Vermaak, J., Doucet, A., Perez, P.: Maintaining multimodality through mixture tracking. In: *ICCV* (2003)
5. Giebel, J., Gavrilu, D., Schnörr, C.: A Bayesian framework for multi-cue 3rd object tracking. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3024, pp. 241–252. Springer, Heidelberg (2004)
6. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
7. Leibe, B., Schindler, K., Van Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. In: *ICCV* (2007)
8. Jiang, H., Fels, S., Little, J.J.: A linear programming approach for multiple object tracking. In: *CVPR* (2007)
9. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: *CVPR* (2008)
10. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: *CVPR* (2008)
11. Berclaz, J., Fleuret, F., Fua, P.: Robust people tracking with global trajectory optimization. In: *CVPR* (2006)
12. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. *Int. J. Comput. Vision* 63, 153–161 (2005)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR* (2005)
14. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *CVPR* (1999)
15. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. *Discrete Appl. Math.* 123, 155–225 (2002)
16. Wolf, J.K., Viterbi, A.M., Dixson, G.S.: Finding the best set of K paths through a trellis with application to multitarget tracking. *IEEE T. Aero. Elec. Sys.* 25 (1989)
17. Rother, C., Kolmogorov, V., Lempitsky, V.S., Szummer, M.: Optimizing binary mrfs via extended roof duality. In: *CVPR* (2007)
18. Miller, E.: Alternative tilings for improved surface area estimates by local counting algorithms. *Comput. Vis Image Und.* 74, 193–211 (1999)
19. Middleton, L., Sivaswamy, J.: *Hexagonal Image Processing: A Practical Approach*. Springer, Heidelberg (2005)
20. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. *IEEE T. Pattern Anal.* 30, 267–282 (2008)
21. Kasturi, R., Goldgof, D.B., Soundararajan, P., Manohar, V., Garofolo, J.S., Bowers, R., Boonstra, M., Korzhova, V.N., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE T. Pattern Anal.* 31, 319–336 (2009)
22. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: *CVPR* (2009)

# Discriminative Spatial Attention for Robust Tracking

Jialue Fan<sup>1</sup>, Ying Wu<sup>1</sup>, and Shengyang Dai<sup>2,\*</sup>

<sup>1</sup> Northwestern University  
2145 Sheridan Road, Evanston, IL 60208

<sup>2</sup> Sony US Research Center  
1730 N. 1st St, San Jose, CA 95112

**Abstract.** A major reason leading to tracking failure is the spatial distractions that exhibit similar visual appearances as the target, because they also generate good matches to the target and thus distract the tracker. It is in general very difficult to handle this situation. In a selective attention tracking paradigm, this paper advocates a new approach of discriminative spatial attention that identifies some special regions on the target, called *attentional regions* (ARs). The ARs show strong discriminative power in their discriminative domains where they do not observe similar things. This paper presents an efficient two-stage method that divides the discriminative domain into a local and a semi-local one. In the local domain, the visual appearance of an attentional region is locally linearized and its discriminative power is closely related to the property of the associated linear manifold, so that a gradient-based search is designed to locate the set of local ARs. Based on that, the set of semi-local ARs are identified through an efficient branch-and-bound procedure that guarantees the optimality. Extensive experiments show that such discriminative spatial attention leads to superior performances in many challenging target tracking tasks.

## 1 Introduction

Our computer vision research on target tracking always aims to develop methods that can work as good as the human. Large research efforts have been devoted to region-based tracking and have produced many outstanding methods, e.g., the mean-shift tracker [17], the kernel-based tracker [3], and the ensemble tracker [19], etc. The major research has been largely focused on effective image region matching to handle large variations in images, and efficient search to locate the target. However, many real applications in video analysis always demand trackers that are more robust and can perform for a longer duration.

Among many reasons that lead to tracking failure, one of the most difficult cases is due to the distractions in the environment that present similar visual appearances as the target and thus exhibiting good matching to the target. These

---

\* This work was conducted while the third author was a graduate student at Northwestern University.

distractions can be from the background clutter or from similar objects in the scene. As the distractions produce false positives in target detection, they lead to wrong association to the tracker, and thus fail the tracker. Because they do give good matches to the target, it is difficult to detect such a distraction failure promptly based on their matching scores.

It is known that our human dynamic visual perception is selective [18], which allows the processing in our visual system to be concentrated on relevant and important visual information. The selection occurs in all stages in visual processing, and it can be based on both innate principles as well as learned heuristics. It is the visual selection that makes our visual system efficient and adaptive in following moving targets. Among many possible kinds of visual selections, spatial attention focuses the computation on some selected local image regions on the target, called *Attentional Regions* or ARs. Tracking the target is fulfilled by the tracking of these ARs. This mechanism appears to be a key in handling clutters, distractions and occlusions in target tracking.

To introduce spatial attention to the design of tracking algorithms, in addition to the matching and searching of ARs, the selection of ARs is a critical issue for persistent tracking. We often observe an interesting phenomenon in various region-based tracking methods that the initialization of the target region may largely influence the tracking performance. A slightly different initialization of the target region sometimes ends up with a much better or worse result. Unfortunately, this phenomenon has not received much attention in the literature, although it conveys a strong message that the selection of ARs cannot be arbitrary. This paper is concerned on finding ARs on the target so as to achieve more robust and persistent tracking.

More specifically, an AR is a local image region that has the largest discriminative power among others in its spatial domain. This spatial selection task is not trivial. For a given target, the number of its candidate attentional regions (i.e., any sub image region on the target) are enormous. Although we can examine all ARs in a brute-force way, we cannot afford its  $O(n^2)$  complexity in practice because  $n$  (i.e., the number of candidates) is huge, and thus a more efficient method is desirable.

This paper presents a novel and efficient solution to the spatial selection of discriminative attentional regions. In the feature space, the feature of an AR has a large *margin* to its nearest neighbors, and we can use this margin in the feature space to represent the discriminative power of an AR. The larger the margin, the more distinctive an AR is in its spatial domain. An AR needs to be distinctive in both its small spatial neighborhood (i.e., local) and a larger domain (i.e., semi-local) that is determined by the possible motion of this attentional region. In the local domain, the local neighbors of an attentional region approximately span a local linear manifold, so that we recast the discriminative power to be a condition number measure of this local linear manifold, and design an efficient gradient-based search for all local ARs. In the semi-local domain, as the approximation does not hold, we design an effective branch-and-bound search that largely reduces the complexity while achieving the optimality. Our extensive

experiments show that the selected discriminative attentional regions are more resilient to distractions and lead to robust tracking.

The novelty of this work includes the following four aspects. (1) Because most existing tracking methods focus on matching but spatial distractions also exhibit good matches, these methods are challenged. This paper explicitly handles the distractions by discovering attentional regions that are resilient to distractions. (2) The proposed approach to locating ARs considers both local and semi-local distractions. This new approach leads to an efficient solution that integrates a gradient-based search and a branch-and-bound search. (3) Based on the spatial selection, this paper presents a new robust tracking algorithm that uses multiple ARs and is adaptive to the appearance changes of the target and the dynamic scene.

## 2 Related Work

In this section, we briefly review recent approaches related to our work. Region-based tracking has been studied in [17,3,5,7,8,13]. In [7], the spatial configuration of the regions is done by optimizing the parameters of a set of regions for a given class of objects. However, this optimization needs to be done off-line. In [8], a method for a well known local maximally stable extremal region (MSER) has been proposed. As the backward tracking is integrated, it restricts its application to off-line tracking.

There is a vast literature on salient region selection [10,15,4,11,12,6,12]. In these works, spatial selection expects the regions to be located at corner-like points. They emphasize the repeatability of the regions in matching. The repeatability of the regions is related to the local discrimination introduced in this paper. But this paper goes one step further. Beside the local discrimination, this paper also studies the semi-local case.

It is worth mentioning that the proposed AR selection mechanism is different from the feature selection paradigms [9]. Feature selection aims to choose global features that best discriminate the object from the background. The target is treated as a whole in those approaches. While in the proposed method, the target is represented by a set of spatial attentional regions. Such a difference in modeling leads to the difference in the selection. In feature selection methods, discriminative features are selected to separate the target and the background, but the AR selection chooses local distinctive image sub-regions (rather than the features). Since the spatial distracters exhibit similar visual appearances as the target, choosing whatever features always results in similar feature vectors. Therefore, feature selection methods are limited in handling this case. On the contrary, the proposed spatial selection method pinpoints to the actual spatial distinctions, and thus is well able to cope with such spatial distracters.

The most closely related work to the proposed method may be [5]. In [5], a general framework of spatial selective attention was advocated for tracking. The early selection process extracts a pool of ARs that are defined as the salient image regions which have good localization properties, and the late selection process dynamically identifies a subset of discriminative attentional regions through

a discriminative learning on the historical data on the fly. However, this work is a large leap from [5], not only because this work presents a much more in-depth study of spatial selection, but also it makes the general selective attentional tracking framework more practical and more effective in practice. The main differences include: (1) The tracking method in [5] is a very specific implementation, and many components in this framework need further investigation and improvement. Moreover, it selects the ARs that are only local discriminative, and it is quite limited in handling the semi-local distraction which is much more common and more challenging in practice. On the contrary, the proposed method selects the ARs that are both local and semi-local discriminative. (2) We explicitly define discriminative margin, which is a new concept, and consider the local discriminative and semi-local discriminative in a unified way. On the contrary, the late selection in [5] is not as principled as the proposed approach.

### 3 Attentional Region (AR)

#### 3.1 Spatial Discrimination

An attentional region (or AR) is a local image region which has the largest discriminative power among others in its spatial domain. At the first step, we need to define a general discriminative measure.

Given a region  $R(\mathbf{x})$  located at position  $\mathbf{x}$  in an image, we denote the set of its neighboring regions by  $\{R(\mathbf{y}), \mathbf{y} \in \mathcal{N}(\mathbf{x})\}$ , where  $\mathcal{N}(\mathbf{x})$  is the spatial neighborhood of  $\mathbf{x}$ , and we call it the *discriminative domain*. The visual features of  $R(\mathbf{x})$  is represented by the feature vector  $\mathbf{f}(\mathbf{x})$ . Denote by  $D(\cdot, \cdot)$  the metric to measure the difference of two feature vectors. Then we define the general discriminative score  $\rho(\mathbf{x})$  of the AR  $R(\mathbf{x})$  by:

$$\rho(\mathbf{x}) \triangleq \min_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})). \quad (1)$$

It is clear that the larger the  $\rho(\mathbf{x})$  is, the more discriminative the AR  $R(\mathbf{x})$  is from its neighbors. If  $\rho(\mathbf{x}) = 0$ , i.e., there is a perfect match in the neighborhood, then this AR has no discriminative power.

However, in practice, we recognize the fact that the most similar one is very likely to be located in a very close vicinity  $\mathcal{L}(\mathbf{x})$ , i.e.,  $\min_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y}))$  is very likely equal to  $\min_{\mathbf{y} \in \mathcal{L}(\mathbf{x})} D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y}))$ . Then this discriminative score can only reflect the local discrimination. To characterize the semi-local discrimination, we should exclude  $\mathcal{L}(\mathbf{x})$  when we define the discriminative score. Let  $\mathcal{S}(\mathbf{x}) = \mathcal{N}(\mathbf{x}) \setminus \mathcal{L}(\mathbf{x})$ . So in practice, we define the discriminative scores  $\rho_S(\mathbf{x})$  and  $\rho_L(\mathbf{x})$  for semi-local and local domains, respectively:

$$\rho_S(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{S}(\mathbf{x})} D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})), \quad (2)$$

$$\rho_L(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{L}(\mathbf{x})} D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})). \quad (3)$$



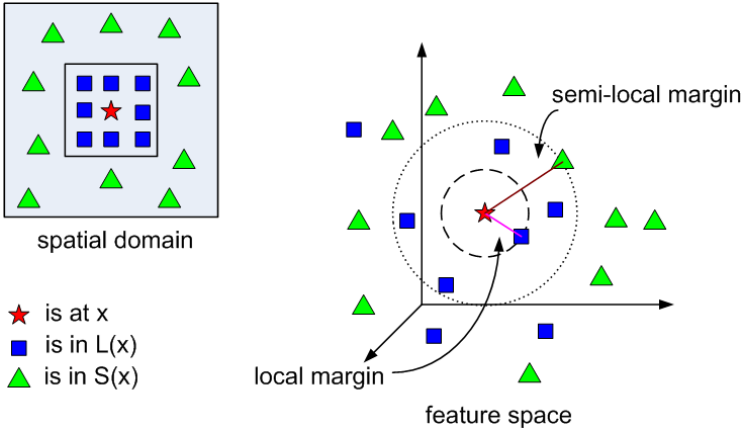


Fig. 1. The discriminative margins for a certain AR

Figure 1 illustrates this concept. In the spatial domain, the red star represents  $\mathbf{x}$ , the blue squares represent some  $\mathbf{y} \in \mathcal{L}(\mathbf{x})$ , and the green triangles represent some  $\mathbf{y} \in \mathcal{S}(\mathbf{x})$ . We also show them in the feature space where the distance between two points is determined by the distance measure  $D(\cdot, \cdot)$ . The hypersphere  $O_L$  is centered at  $\mathbf{x}$  with a radius  $\rho_L(\mathbf{x})$ . Therefore, all the blue squares are out of the hypersphere, and there is at least one blue square on the boundary of the hypersphere. It is clear that the discriminative score  $\rho_L(\mathbf{x})$  reflects the *margin* between the target and the set of its local neighbors in the feature space. The larger the  $\rho_L(\mathbf{x})$  is, the more local discriminative the AR  $R(\mathbf{x})$  is. Similarly, the hypersphere  $O_S$  is centered at  $\mathbf{x}$  with the radius  $\rho_S(\mathbf{x})$ . The discriminative score  $\rho_S(\mathbf{x})$  reflects the *margin* between the target and the set of its nearest semi-local neighbors in the feature space.

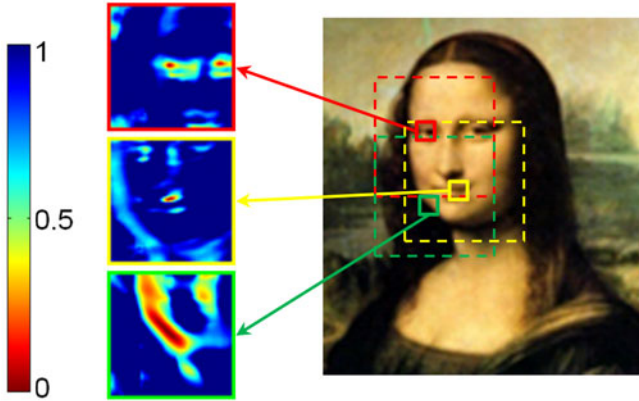
### 3.2 Attentional Region

An AR needs to be distinctive in both its local spatial neighborhood (i.e., the local domain) and a larger domain (i.e., the semi-local domain).

We denote the set of *local* ARs by  $\mathcal{X}_L = \{\mathbf{x} : \rho_L(\mathbf{x}) > \epsilon_L\}$  where  $\epsilon_L > 0$  is a threshold for the local domain. Similarly, denote the set of *semi-local* ARs by  $\mathcal{X}_S = \{\mathbf{x} : \rho_S(\mathbf{x}) > \epsilon_S\}$ . By definition, an AR needs to be discriminative at both local and semi-local domains. Therefore, the set of ARs  $\mathcal{X} = \mathcal{X}_L \cap \mathcal{X}_S$ .

The intuitive explanation of the difference between AR and a common region is shown in Fig. 2. In Fig. 2, three representative patches are chosen, and the matching scores between the selected patches and their neighbors are visualized.

As shown in Fig. 2, the matching error surfaces of the AR and the common regions behave quite differently: The region at the chin has a poor local discriminative power since its neighbors along the boundary looks quite similar. The region at the eye has a poor semi-local discriminative power, because there is a similar eye corner in the valid semi-local domain and it acts as the distractor.



**Fig. 2.** Three regions and their matching error surfaces with their corresponding neighbor regions



**Fig. 3.** ARs are related to their associated discriminative domain. The leftmost is the local ARs. When  $\mathcal{N}(\mathbf{x})$  becomes larger, there exists a less number of ARs. As shown in the rightmost, only three ARs survive in the largest range we specified. The positions are at the mouth and the joint part between the leg and the body of the zebra.

The region at the mouth has both strong semi-local and local discriminative power as good matches are only focused in a very small neighborhood. Traditional methods [10, 4, 11, 12] may examine those local ARs but are unable to identify the semi-local ones, because they only consider the local properties.

Whether a region is discriminative or not is related to the range of the associated discriminative domain  $\mathcal{N}(\mathbf{x})$ . A region is an AR in a spatial domain if and only if there are no distractors (i.e., good matches) in this domain. When the domain becomes larger, some distractors may be present, and thus reduce the discriminative power of this region in the larger domain. If the discriminative power becomes below the threshold, this region is no longer an AR. Thus, when we keep enlarging the discriminative domain, we have fewer and fewer ARs. Figure 3 shows one example to illustrate this situation.

### 4 Spatial Selection of ARs

For a given target, denote the set of its candidate regions (i.e., any sub image region on the target) by  $\mathcal{A}$ . The spatial selection task, i.e., finding the ARs in  $\mathcal{A}$ , is not trivial. Comparing all regions with all of their neighbors in a brute-force

way is computationally infeasible, because the number of the candidate regions is huge, and thus a more efficient method is needed.

We propose a two-step method to find ARs: (1) we first obtain all local ARs  $\mathcal{X}_L$  based on an efficient gradient-based search. (2) Then we select a subset of  $\mathcal{X}_L$ , whose element has strong semi-local discriminative power to be ARs through an efficient branch-and-bound search that guarantees the optimality.

### 4.1 Gradient-Based Search for Local ARs

For a region located at  $\mathbf{x}$ , assume  $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^d$ . The visual features of its local spatial neighbors constitute a linear manifold (up to two dimensional) at  $\mathbf{f}(\mathbf{x})$  in the feature space. Assume  $\Delta\mathbf{x} = [\Delta u, \Delta v]^T$ , we have

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \Phi \Delta\mathbf{x}, \tag{4}$$

where  $\Phi \triangleq [\frac{\partial \mathbf{f}}{\partial u} \ \frac{\partial \mathbf{f}}{\partial v}]$  is a  $d \times 2$  matrix.

Using L2 metric for matching, the local discriminative margin  $\rho_L(\mathbf{x})$  becomes:

$$\rho_L(\mathbf{x})^2 = \min_{\mathbf{x} + \Delta\mathbf{x} \in \mathcal{L}(\mathbf{x})} \|\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) - \mathbf{f}(\mathbf{x})\|^2 \approx \min_{\mathbf{x} + \Delta\mathbf{x} \in \mathcal{L}(\mathbf{x})} (\Delta\mathbf{x})^T \mathbf{A} \Delta\mathbf{x}, \tag{5}$$

where  $\mathbf{A} \triangleq \Phi^T \Phi$  is a  $2 \times 2$  matrix which characterizes this local linear manifold.

**Case 1:**  $rank(\mathbf{A}) = 1$ . It is clear that  $\rho_L(\mathbf{x}) = 0$ .

**Case 2:**  $rank(\mathbf{A}) = 2$ . The minimum is obtained at the inner boundary of  $\mathcal{L}(\mathbf{x})$  due to the discretization of  $\mathbf{x}$ . Assume the inner boundary of  $\mathcal{L}(\mathbf{x})$  to be  $\|\Delta\mathbf{x}\| = 1$ . Then we have

$$\rho_L(\mathbf{x})^2 = \min_{\|\Delta\mathbf{x}\|=1} (\Delta\mathbf{x})^T \mathbf{A} \Delta\mathbf{x}. \tag{6}$$

We perform SVD on  $\Phi$  and obtain two singular values  $\sigma_1$  and  $\sigma_2$ . Without loss of generality, we assume  $\sigma_1 \geq \sigma_2$ . As  $\mathbf{A} = \Phi^T \Phi$ ,  $\sigma_1^2$  and  $\sigma_2^2$  are the eigenvalues of  $\mathbf{A}$ .

We can easily see that  $\rho_L(\mathbf{x})^2 = \sigma_2^2$ . Therefore, maximizing the margin  $\rho_L(\mathbf{x})$  is equivalent to maximizing  $\sigma_2$ . It is clear that when  $det(\mathbf{A})$  becomes larger,  $\rho_L(\mathbf{x})$  will become larger, and then the problem becomes meaningless. But considering the fact that  $det(\mathbf{A})$  is bounded, i.e.,  $det(\mathbf{A}) \leq \chi^2$ , and the fact that  $det(\mathbf{A}) = (\sigma_1 \sigma_2)^2$ , we have

$$\sigma_2^2 = \chi \frac{\sigma_2^2}{\chi} \leq \chi \frac{\sigma_2^2}{\sigma_1 \sigma_2} = \chi \frac{1}{\sigma_1 / \sigma_2}. \tag{7}$$

It is clear that maximizing  $\sigma_2$  amounts to minimizing the condition number  $\sigma_1 / \sigma_2$  of  $\Phi$ .

The above analysis reveals the relation between the discriminative power of a region and the singularity property of its local linear manifold.

In practice, only obtaining the criterion for local AR placement is insufficient, since it is not attractive to exhaustively evaluate this criterion all over the image.

In [4], a gradient descent algorithm has been proposed to efficiently find good placement where the condition number of  $\Phi^T \Phi$  is locally minimized. We follow that algorithm in this paper. First we randomly initialize a set of AR candidates. Following the gradient of the condition number these ARs converge to their corresponding local minima. The set of local minima is  $\mathcal{X}_L$ .

The matrix  $\Phi$  depends on the choices of the feature space and the matching metric. In this paper we use the contextual flow [16] as the feature vector, because it is robust to small changes on local appearance that invalidate the constancy in brightness.<sup>1</sup>

## 4.2 Branch-and-Bound Selection of ARs from $\mathcal{X}_L$

Based on the set of local ARs  $\mathcal{X}_L$  obtained in Sect. 4.1, we obtain ARs with a strong semi-local discriminative power from  $\mathcal{X}_L$ . We solve a more general and flexible problem as follows:

Given the set  $\mathcal{X}_L = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  (i.e.,  $|\mathcal{X}_L| = N$ ), we want to choose the ARs  $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_M \in \mathcal{X}_L$  with the  $M$  largest discriminative score  $\rho_S(\cdot)$ .

Since the linear approximation is invalid in the semi-local discriminative domain  $\mathcal{S}(\mathbf{x})$ , differential approaches are not appropriate. A brute-force exhaustive method is:  $\forall \mathbf{x} \in \mathcal{X}_L$ , we calculate  $\rho_S(\mathbf{x})$ , and then select the most discriminative ones. The complexity is  $O(|\mathcal{S}(\mathbf{x})| \times N)$ , and is still intensive in practice.

Here we use a branch-and-bound search which largely reduces the complexity while maintaining the same optimal result as by the exhaustive search.

Let  $\mathcal{S}(\mathbf{x}) = \{\mathbf{x} + \Delta \mathbf{l}_1, \dots, \mathbf{x} + \Delta \mathbf{l}_n\}$ , where  $n = |\mathcal{S}(\mathbf{x})|$ , and  $\Delta \mathbf{l}_i$  is the relative position between the target AR and its  $i$ th neighbor in the semi-local discriminative domain. Denote  $\rho_i(\mathbf{x}) = \min_{\mathbf{y} \in \{\mathbf{x} + \Delta \mathbf{l}_1, \dots, \mathbf{x} + \Delta \mathbf{l}_i\}} D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y}))$ . Then we have  $\rho_i(\mathbf{x}) = \min\{\rho_{i-1}(\mathbf{x}), D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x} + \Delta \mathbf{l}_i))\}$ , thus  $\rho_1(\mathbf{x}) \geq \dots \geq \rho_n(\mathbf{x}) = \rho_S(\mathbf{x})$ . In the beginning, we initialize an empty priority queue  $P$  to store the candidates. For each  $\mathbf{x}_i \in \mathcal{X}_L$ , we calculate  $\hat{\rho}(\mathbf{x}_i) = \rho_1(\mathbf{x}_i)$  as the upper bound of  $\rho_S(\mathbf{x}_i)$ . Then we sort  $\{\hat{\rho}(\mathbf{x}_i)\}$  in the descending order and push them sequentially into  $P$  so that the top state has the largest  $\hat{\rho}(\cdot)$ . For each  $\mathbf{x}$ , we associate a variable  $\gamma(\mathbf{x})$  to count the number of elements in  $\mathcal{S}(\mathbf{x})$  which has been searched around  $\mathbf{x}$ .

At every iteration, we retrieve the top state  $(\mathbf{x}, \hat{\rho}(\mathbf{x}))$  from  $P$ , where  $\hat{\rho}(\mathbf{x})$  is the current upper bound of  $\rho_S(\mathbf{x})$ , and  $\hat{\rho}(\mathbf{x}) = \rho_{\gamma(\mathbf{x})}(\mathbf{x})$ . If  $\gamma(\mathbf{x}) = n$ , meaning that we have already sought all the neighbors in  $\mathcal{S}(\mathbf{x})$ , we output  $\mathbf{x}$  into the set of ARs and remove  $\mathbf{x}$  from  $P$ .

Otherwise  $\gamma(\mathbf{x}) < n$ , we increase  $\gamma(\mathbf{x})$  by 1, calculate  $\mathbf{f}(\mathbf{x} + \Delta \mathbf{l}_{\gamma(\mathbf{x})})$ , and update the upper bound

$$\hat{\rho}(\mathbf{x}) := \min\{\hat{\rho}(\mathbf{x}), D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x} + \Delta \mathbf{l}_{\gamma(\mathbf{x})}))\}. \quad (8)$$

Then we insert  $(\mathbf{x}, \hat{\rho}(\mathbf{x}))$  into  $P$  maintaining the property that  $P$  is sorted with the descending order of  $\rho(\cdot)$  (replace the old  $\mathbf{x}$ ). Then we retrieve the top state

<sup>1</sup> In [16],  $\Phi$  is the contextual gradient and can be computed directly in a closed form.

**Table 1.** The branch-and-bound algorithm for selecting ARs

---

**Input**  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \Delta\mathbf{l}_1, \dots, \Delta\mathbf{l}_n$

**Output**  $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_M$

1. FOR  $i = 1$  TO  $N$  DO
  - calculate  $\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_i + \Delta\mathbf{l}_1)$ ,
  - set  $\hat{\rho}(\mathbf{x}_i) = D(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_i + \Delta\mathbf{l}_1))$ ,
  - $\gamma(\mathbf{x}_i) = 1$
- Initialize  $P$  as empty priority queue.  $c = 0$ .
2. Sort  $\{\hat{\rho}(\mathbf{x}_i)\}$  in descending order.
  - Let  $\hat{\rho}(\bar{\mathbf{x}}_1) \geq \dots \geq \hat{\rho}(\bar{\mathbf{x}}_N)$ ,
  - FOR  $i = N$  TO  $1$  DO
    - push  $(\bar{\mathbf{x}}_i, \hat{\rho}(\bar{\mathbf{x}}_i))$  into  $P$
3. Retrieve top state  $(\mathbf{x}, \hat{\rho}(\mathbf{x}))$  from  $P$ .
4. If  $\gamma(\mathbf{x}) = n$ 
  - $c = c + 1, \hat{\mathbf{x}}_c = \mathbf{x}$ , goto 3.
  - Else goto 5.
5. If  $c = M$ , Return. Else goto 6.
6.  $\gamma(\mathbf{x}) = \gamma(\mathbf{x}) + 1$ 
  - Calculate  $\mathbf{f}(\mathbf{x} + \Delta\mathbf{l}_{\gamma(\mathbf{x})})$
  - Set  $\hat{\rho}(\mathbf{x}) = \min\{\hat{\rho}(\mathbf{x}), D(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x} + \Delta\mathbf{l}_{\gamma(\mathbf{x})}))\}$
  - Insert  $(\mathbf{x}, \hat{\rho}(\mathbf{x}))$  into  $P$  so that  $P$  is still sorted w.r.t.  $\hat{\rho}(\cdot)$ . Goto 3.

---

again iteratively until a number of  $M$  ARs are found. The algorithm is summarized in Table 1.

The top state  $\mathbf{x}$  of  $P$  has the largest upper bound of  $\rho_S(\mathbf{x})$ , because for the remaining  $\mathbf{x}_i$ s in  $P$ ,  $\rho_S(\mathbf{x}_i)$  is bounded by  $\hat{\rho}(\mathbf{x})$ . As each time we only consider the most promising  $\mathbf{x}$  of  $P$ , this significantly reduce the complexity. The complexity is  $O(\sum_{i=1}^N \gamma(\mathbf{x}_i))$ , and this method guarantees the optimality.

In practice, the complexity versus the exhaustive search is measured by the ratio  $r = \frac{1}{nN} \sum_{i=1}^N \gamma(\mathbf{x}_i)$ . The value of  $r$  is 0.18 on average for our testing sequences, e.g., for sequence **zebra**,  $r = 0.18$ . For sequence **dolphin**,  $r = 0.16$ . This means that our method significantly reduces the complexity in searching for ARs. Extra operations in our method (*i.e.*, insertion and sorting) have little computational complexity, as those operations take much less time than computing  $D$ .

## 5 Discriminative Attentional Visual Tracking

As the ARs are not similar to the other regions in their discriminative domain, the tracking performance of ARs is very robust. We propose a new attentional tracking method by using AR, and it has three important steps: At the first step, we extract ARs from images. Secondly, the contextual flow tracking algorithm [16] is applied to track each ARs independently. Finally, the beliefs of all the ARs are fused to determine the target location.

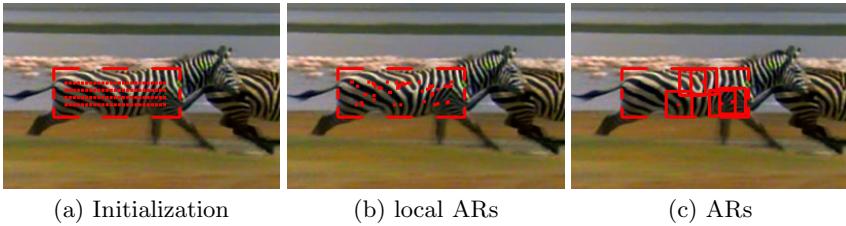


Fig. 4. AR selection

### 5.1 AR Selection/Tracking

At the first frame, the target is initialized by the user. We evenly initialize  $N_{max}$  tentative ARs inside the target (Fig. 4(a)). The local ARs are shown in Fig. 4(b). Figure 4(c) shows top five ARs. For each AR, we record the geometrical relation between the ARs and the target (the relative position and the scale).

For each AR, the tracking is done based on the contextual flow method [16].

### 5.2 Attentional Fusion and Target Estimation

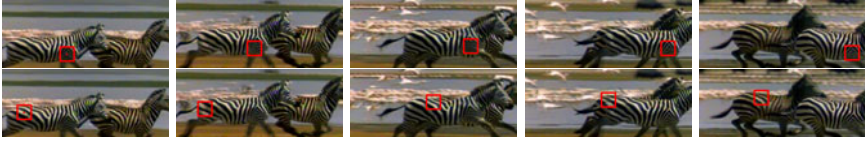
After obtaining the motion of each AR, we apply a Hough-voting scheme [14] to estimate the target location based on the matching scores of ARs and the recorded geometry. The estimated AR location casts a probabilistic vote about the target centroid position with respect to the AR center. The better the matching performance of a certain AR, the higher the probabilistic score. After the votes from all ARs are aggregated into a Hough image, the target location can be estimated as the peak in this image. This scheme is appropriate to handle occlusion. If some ARs are occluded, their matching scores will be very low, thus the probabilistic votes from those ARs are very low, and contribute less to the object location prediction than the ARs which are not occluded.

The scale of the target is estimated by a voting-like approach based on the scale estimation for each AR. To obtain a robust estimation, we only count the ARs which have high matching scores.

### 5.3 Model Adaptation

As the appearance changes, due to view differences, illumination variations and shape deformation, can ruin the observation, the model adaptation mechanism is necessary. We adapt the model by updating the ARs when necessary. The matching score of each AR measures the variation of its appearance. If the matching performance is good enough, we call the AR *active*. Otherwise, for a certain AR, if the matching score has been low for a long period of time (e.g., consecutive 10 frames), we call it *inactive* since it probably undergoes appearance changes or short term occlusion.

At the current frame, after target estimation, we check the matching score for each AR to see if it remains active. When there are  $m$  inactive ARs at the current frame, we remove them and select  $m$  new ARs from the target.



**Fig. 5.** Comparison of different placement of one AR. (Top) the AR from our method (bottom) the local AR

## 6 Experiments

For tracking initialization, we evenly initialize  $N_{max} = 100$  tentative ARs inside the target. The size of the ARs is  $25 \times 25$ . For a certain AR, the size of its discriminative domain  $\mathcal{N}(\mathbf{x})$  is determined by its possible motion and the maximum search range for tracking. The larger the possible motion, the larger  $\mathcal{N}(\mathbf{x})$  we use.

Without code optimization, our C++ implementation comfortably runs at around 15 fps on average on Pentium 3G for  $320 \times 240$  images.

We compare our method with an attentional visual tracker (AVT) [5] that reported excellent tracking performance. For fair comparison, we use the contextual flow as the feature vector, and use the Hough voting scheme in the fusion process for both methods. In addition, we have included the late selection procedure in AVT for comparison.

### 6.1 Using the Most Discriminative AR

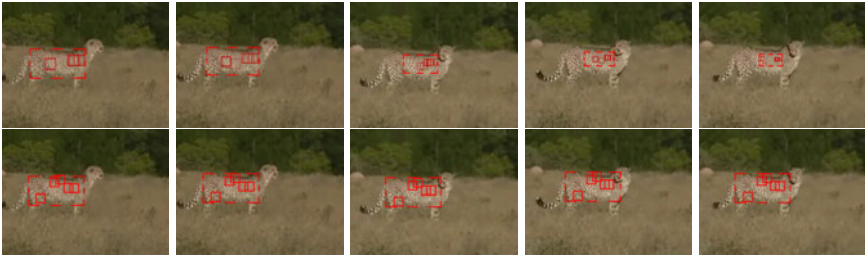
ARs are resilient to distractors, because by definition an AR is not confused by its neighboring regions in its discriminative range. In this experiment, we compare the tracking performance by selecting different ARs and demonstrate the effectiveness of our method. The AR with the largest discriminative power is shown at the top row of Fig. 5. We choose some local ARs for comparison (one example is shown at the bottom row of Fig. 5). It is observed that at the top row, the texture of the best AR is quite different from its neighborhood. While at the bottom row, the texture of the AR contains stripes which is not quite discriminative in its semi-local discriminative domain. Therefore, the tracking performance shows that the local AR is unstable during tracking (keep drifting) while the AR at the top row succeeds and is very stable.

### 6.2 Handling Local Appearance Changes

Tracking targets undergoing local deformation is difficult in practice. However, if the local deformation only occurs in some parts of the target, the ARs on other parts can still make the tracking robust. These stable ARs contribute more in the fusion process as they have strong matching, so the tracking performance is still good. The comparison result is shown in Fig. 6 and 7. In Fig. 6, although the target appearance changes at some parts, the bottom-right AR is persistently



**Fig. 6.** A comparison of DAVT and AVT [dancing]. (Top) AVT (bottom) the proposed method



**Fig. 7.** A Comparison of DAVT and AVT [cheetah]. (Top) AVT (bottom) the proposed method

robust and thus dominates the fusion and gives good tracking results. The white ARs indicate those that have relative bad matching. Although these white ARs sometimes do not have strong matching, in most cases they are robust, since they are located at the boundary of the face and there are no distractors nearby. In Fig. 7, the textures of the cheetah are very similar. The ARs found by the proposed method are near the back and thigh of the cheetah. These regions look different from the body of the cheetah, so they hardly drift to some other regions inside the body. However, for AVT, it only selects some local ARs. We observe that there are some distractors in the semi-local domain of these local ARs and AVT fails as shown in Fig. 7.

We manually labeled the ground truth of our testing sequences to evaluate the tracking performance. Figure 8 shows the comparison of tracking error over time on the `bicycle` sequence (we use a different initialization as in 5). At the 330th Frame, AVT is distracted and fails, but our method keeps the track persistently.

### 6.3 Handling Scale, Rotation and Occlusion

The scale estimation can be handled since the selected ARs are stable and rarely distracted. As in our matching method, the contextual descriptor is rotation invariant if we only use color contexts, the ARs give accurate matching despite of the motion. Then we estimate the rotation by measuring the relative position



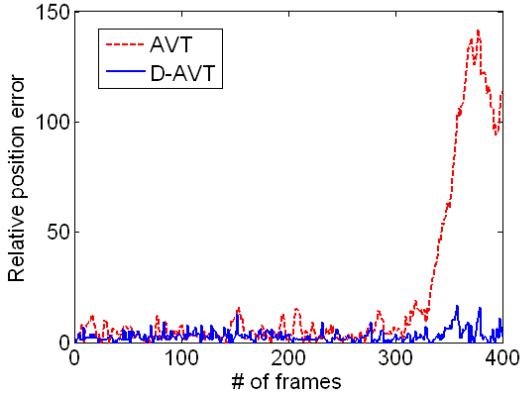


Fig. 8. Comparison: tracking errors between DAVT and AVT

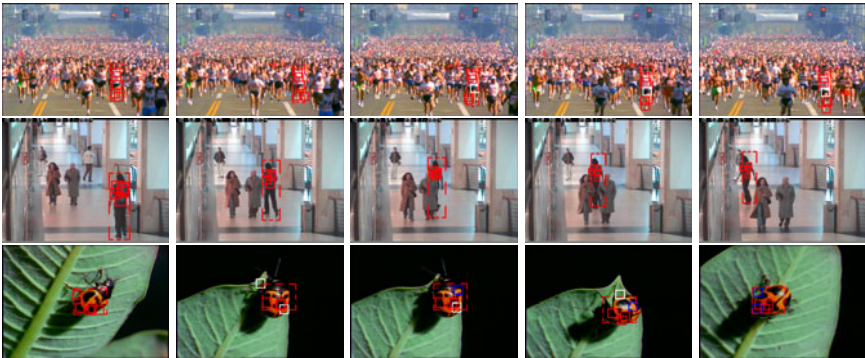


Fig. 9. Three examples of the proposed method

between the ARs. The occlusion can be handled by the fusion process. The model adaptation is also illustrated. Three examples are shown in Fig. 9. On the bottom row, the blue ARs indicate those that have been updated.

## 7 Conclusion

Spatial distraction is a major culprit for tracking failure, because distractors also exhibit good matching. This paper presents a novel approach of discriminative spatial attention to overcome this challenge, by selecting a set of discriminative attentional regions on the target. The discrimination power of an attentional region is defined by the margin of its feature from that of those in its discriminative domain. By integrating local and semi-local discrimination, this paper proposes an efficient method in finding ARs. Extensive tests demonstrate that the proposed discriminative spatial attention scheme significantly improves the

robustness in tracking. The further analysis [20] reveals that the existing works on local saliency detection share the common purpose of achieving good localization properties. Therefore, our AR selection scheme is very flexible so that those methods can be alternatively adopted for finding local ARs.

**Acknowledgement.** This work was supported in part by National Science Foundation grant IIS-0347877, IIS-0916607, and US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504.

## References

1. Peters, R., Itti, L.: Beyond bottom-up: incorporating task-dependent influences into a computational model of spatial attention. In: CVPR (2007)
2. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on PAMI*, 1254–1259 (1998)
3. Hager, G., Dewan, M., Stewart, C.: Multiple kernel tracking with ssd. In: CVPR (2004)
4. Fan, Z., Yang, M., Wu, Y., Hua, G., Yu, T.: Efficient optimal kernel placement for reliable visual tracking. In: CVPR (2006)
5. Yang, M., Yuan, J., Wu, Y.: Spatial selection for attentional visual tracking. In: CVPR (2007)
6. Gao, D., Vasconcelos, N.: Discriminant interest points are stable. In: CVPR (2007)
7. Parameswaran, V., Ramesh, V., Zoghلامي, I.: Tunable kernels for tracking. In: CVPR (2006)
8. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (mser) tracking. In: CVPR (2006)
9. Collins, R., Liu, Y.: On-line selection of discriminative tracking features. In: ICCV (2003)
10. Shi, J., Tomasi, C.: Good features to track. In: CVPR (1994)
11. Kadir, T., Brandy, M.: Saliency, scale and image description. *IJCV* (2001)
12. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. *IJCV* (2005)
13. Kwon, J., Lee, K.: Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In: CVPR (2009)
14. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* (2004)
16. Wu, Y., Fan, J.: Contextual flow. In: CVPR (2009)
17. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR (2000)
18. Palmer, S.: *Vision science: photons to phenomenology*. The MIT Press, Cambridge (1999)
19. Avidan, S.: Ensemble tracking. In: CVPR (2005)
20. Fan, J., Wu, Y., Dai, S.: Discriminative spatial attention for robust tracking. Northwestern University Technical Report (2010)

# Object, Scene and Actions: Combining Multiple Features for Human Action Recognition

Nazli Ikizler-Cinbis and Stan Sclaroff

Department of Computer Science, Boston University

**Abstract.** In many cases, human actions can be identified not only by the singular observation of the human body in motion, but also properties of the surrounding scene and the related objects. In this paper, we look into this problem and propose an approach for human action recognition that integrates multiple feature channels from several entities such as objects, scenes and people. We formulate the problem in a multiple instance learning (MIL) framework, based on multiple feature channels. By using a discriminative approach, we join multiple feature channels embedded to the MIL space. Our experiments over the large YouTube dataset show that scene and object information can be used to complement person features for human action recognition.

## 1 Introduction

Action recognition “in the wild” is often a very difficult problem for computer vision. When the camera is non-stationary, and the background is fairly complicated, it is often difficult to infer the foreground features and the complex dynamics that are related to an action. Moreover, motion blur, serious occlusions and low resolution present additional challenges that cause the extracted features to be largely noisy.

Under such challenges, we argue that the features of the scene and/or moving objects can be used to complement features extracted from people in the video. The intuition behind this is straightforward: the presence (or absence) of particular objects or scene properties can often be used to infer the possible subset of actions that can take place. For example, if there is a pool within the scene, then “diving” becomes a possible action. On the contrary, if there is no pool, but a basketball court, then the probability of the “diving” action reduces. In this work, our aim is to capture such relationships between objects, scenes and actions.

Our approach starts with extracting a large set of features for describing both the shape and the motion information in the videos. All the features are extracted densely, allowing spatial and temporal overlap, and we operate over tracks when the temporal continuity is available. We do not use any explicit object detectors, but treat each moving region as an object candidate. In the end, the videos are represented with multiple feature vectors acquired from different feature channels.

We are particularly interested in human action classification in the real-world, i.e. in unconstrained video sources like YouTube. In this problem, the videos are weakly annotated; there is a class label for each video sequence, however we do not know where or when in the video sequence the action occurs. Moreover, there may be more

than one person or moving object in the video, and only a subset of the detected regions are involved in the action. Our aim is to be able to train our action models in the presence of such diverse conditions.

For this purpose, we formulate our problem within a multiple instance learning (MIL) framework, where the training set is ambiguous and the training labels are associated with bags of instances, rather than single instances as in a fully supervised system. The obvious advantage of using such an approach is to tolerate the large amount of irrelevant instances or false detections in the input videos.

In order to accommodate multiple heterogeneous feature types, we define an agglomerative multiple instance learning framework, where each video is represented with multiple bags and each bag corresponds to a different feature channel. The MIL positivity constraint on a bag is therefore extended over multiple bags, i.e., at least one bag is required to contain one positive instance for the particular action. We then formulate a discriminative learning strategy with globally weighted or unweighted combinations of these multiple bags. We test our approach over the extensive YouTube dataset provided by Liu et al. [22], and the results demonstrate that the proposed framework effectively combines different and noisy feature channels for accurate human action recognition.

## 2 Related Work

Human action recognition has been a very active research topic over the recent years. This makes the comprehensive listing of the related literature impossible, while Forsyth et al. [10] presents an extensive review of the subject. Some of the recent works include [8,29,18,20,16,13]. In most of the earlier works, the focus is on simpler scenarios, where the background was stable and the foreground human figure is easy to extract [4,18]. However, this scenario is hardly realistic; videos from the real world are fairly complicated, especially when taken in uncontrolled environments. Some recent approaches try to deal with such complex scenarios [19,25,20,16].

Joint modeling of object and action interactions has been a recent topic of interest. Moore et al.'s work [26] is one of the earliest attempts to consider actions and objects together. They use belief networks for modeling object and hand movements extracted from static camera sequences. Gupta et al. [12] try to improve the localization of both objects and actions by using a graphical Bayesian model. Marszalek, et al. [24] use movie scripts as automatic supervision for scene and action recognition in movies. Han et al. [13] use context and higher level bags-of-detection descriptors for action recognition. They assume that the objects related to each action are known beforehand and corresponding object detectors are available. In our case, we do not rely on explicit object detectors and try to discover related objects in an unsupervised manner. We consider each moving region as a candidate object region and we utilize shape and motion descriptors for all candidate regions. While doing this, we have no explicit knowledge about their class membership.

Multiple Instance Learning (MIL) paradigm has been explored in quite a number of studies, both in machine learning [2] and computer vision [23,31,5]. Computer vision problems are in fact very suitable application domains for MIL algorithms, because of the high-level of ambiguity in the domain. There are also some recent works

which use multiple instance learning for tracking and human actions. Babenko et al. [3] introduce an online MIL algorithm for target tracking. Ali and Shah use multiple-instance embedding [1] to facilitate classification with their kinematic mode features. Hu et al. [14] utilize a simulated annealing based MIL algorithm for finding the exact location of the actions over HOG features.

YouTube videos have been a focus of interest recently, due to its popularity being a widespread source that contains various challenging videos. Niebles, et al. [27] present a method for detecting moving people from such videos. Ikizler-Cinbis, et al. [17] use web images to facilitate action recognition in uncontrolled videos. Tran, et al. [30] work on YouTube Badminton videos. Recently, Liu, et al. [22] collected a large action dataset, and presented a method based on PageRank algorithm to prune the large number of space-time interest points in these videos.

### 3 Features

Features constitute the basic building blocks of our algorithm. Here, the idea is to extract as many meaningful and informative features as possible, both at the high and low-level. These features are extracted densely, in the sense that there can be spatial or temporal overlap between them. We will rely on the learning algorithm to extract useful patterns that associate each action with the combination of these different sets of features. There are three sets of features, namely “person-centric”, “object-centric” and “scene-centric” features. All these feature channels are depicted in Fig. 1. In this section, we first describe the video pre-processing steps and then go into the details of the feature extraction procedure.

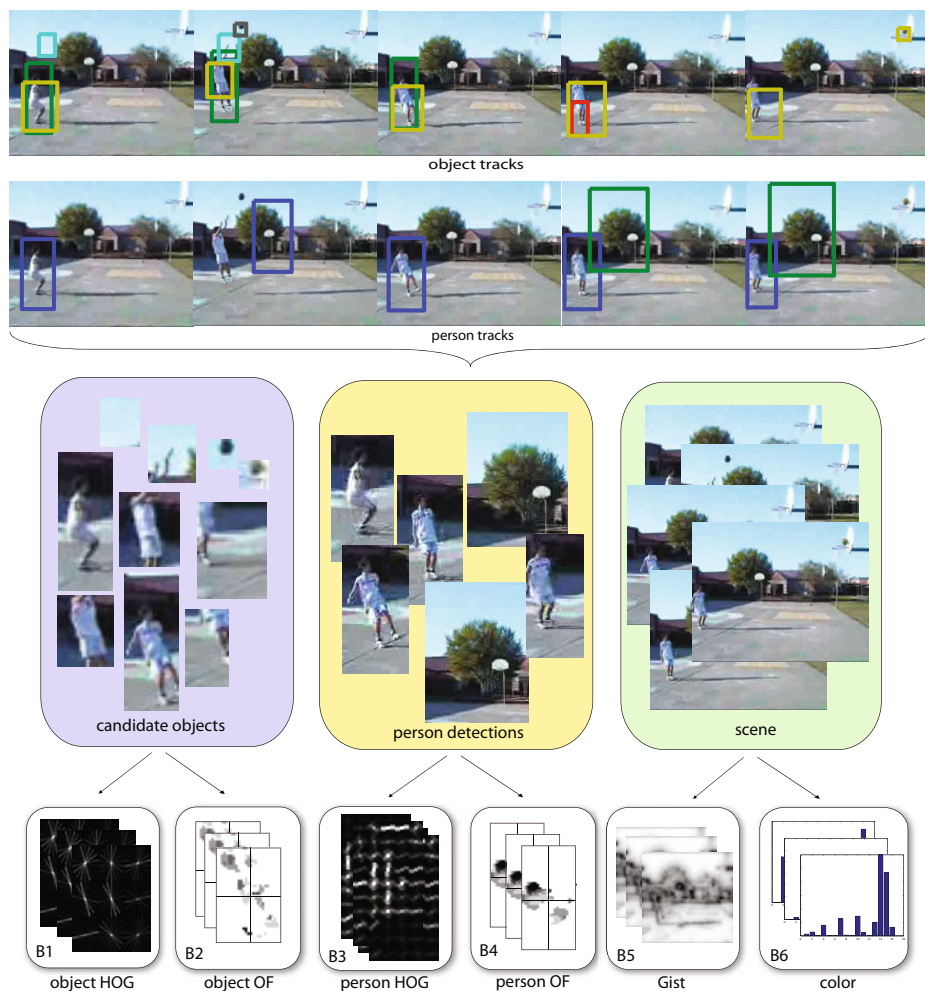
#### 3.1 Stabilizing Videos

When there is camera motion and the background is not static, the optical flow of the foreground objects is not easy to estimate, amidst the noisy flow field. Therefore, before extraction of features, especially the motion-related ones, the videos should be stabilized. We use a dominant motion compensation procedure for this purpose.

In order to estimate the foreground flow field, we make use of a homography-based motion compensation approach, similar to [21]. Assuming that the background is relatively dominant in the scene, we can estimate the background flow by calculating the homography between consecutive frames. For this purpose, we first extract Harris corner features from each frame. By establishing feature correspondences between frames, we estimate the homography using RANSAC. Once the homography between consecutive frames is estimated, it can be used for calculating both the background flow vector per pixel  $\mathbf{m}_b(x, y)$  and as a prior to a block-based optical flow algorithm for computing the overall flow  $\mathbf{m}_o(x, y)$ . Then, the foreground flow at each pixel  $(x, y)$  is calculated as

$$\mathbf{m}_f(x, y) = (\mathbf{m}_o(x, y) - \mathbf{m}_b(x, y)). \quad (1)$$

The noisy motion flow fields are mostly stabilized by this procedure. Example resultant flows can be seen in Fig. 2.



**Fig. 1.** There are three main feature channels, namely person-centric, object-centric and scene features. Once the videos are stabilized, we extract candidate person and object tracks. An example track from a basketball sequence is shown above (for details of track extraction, see the text). From each track, we extract multiple features. Each of the feature channels may contain noisy detections, as well as the true detections. There can be multiple people and multiple objects within the video. Since we do not have explicit supervision on which feature or detection region may be relevant, each feature channel is defined as a MIL bag. We then combine these feature channels using two different approaches.



**Fig. 2.** The videos include an extensive amount of camera motion, thus, uneven flow fields. We use a homography based approach to estimate the flow of the foreground objects, following [21]. (a) shows the original frames from four videos. (b) shows the flow estimate (in green) without the motion stabilization. (c) shows the foreground flow estimate obtained following stabilization. As seen, this estimate gives us the ability to concentrate on the moving foreground objects.

### 3.2 Person-Centric Features

To extract person-centric features, first, one should have a rough estimate of the location(s) of the person(s) and corresponding person tracks in the video. This is tough, especially when the background clutter is dominant.

We approach the problem by using a “tracking-by-detection” method and use Felzenswalb et al.’s human detector [9]. This person detector has shown to perform quite well in detecting people of various poses; however, due to motion blur and pose variations, it is not able to locate the person in every frame. In order to compensate for this, we use mean-shift tracking [6] to fill the gaps in which the person detector did not fire, by using the person detection bounding box to initiate the tracker in each case. We initiate a separate track for each individual person detection and discard the short tracks (with  $\leq 5$  frames) as being noise.

Figure 1 shows some example tracks. Here, we define a track as the series of bounding boxes associated with the detected regions over the video. While the final tracks are not perfect and some of the tracks may still be irrelevant, they provide fairly usable person localizations. From each detected track, we extract two types of features: person-centric motion and shape features.

**Person-centric motion features:** Optical flow has been shown to be a useful feature for describing human actions [8]. We use the intersection of the estimated foreground flow (computed by stabilizing the video as in Sec. 3.1) and the person tracks to locate the regions of the optical flow map that belong to a person. We describe the flow in each detection bounding box with a spatial histogram, by dividing the optical flow field equally into  $2 \times 2$  spatial regions, and represent each spatial bin with four major flow orientations. In order to accommodate for the noise in the optical flow, we use a windowing scheme over the tracks and extract histograms from every snippet of six frames.

Each subwindow is considered as an instance in the MIL setting (see Section 4). The final descriptor of each instance has  $4 \times 4 \times 6 = 96$  feature dimensions.

**Person-centric shape features:** We expect the shape feature to be complimentary to motion features, especially when the motion field of the video has excessive noise. In order to account for this shape information, we use the Histogram of Oriented Gradients (HOG) [7]. We downsize each bounding box region to  $[64 \times 32]$  pixels and extract HOGs using 8 pixel cell size and 8 pixel cell step. We use eight orientation bins and use a temporal window of five frames over the tracks to accumulate the temporal pattern. The final descriptor has  $8 \times 8 \times 4 \times 5 = 1280$  feature dimensions.

### 3.3 Object-Centric Features

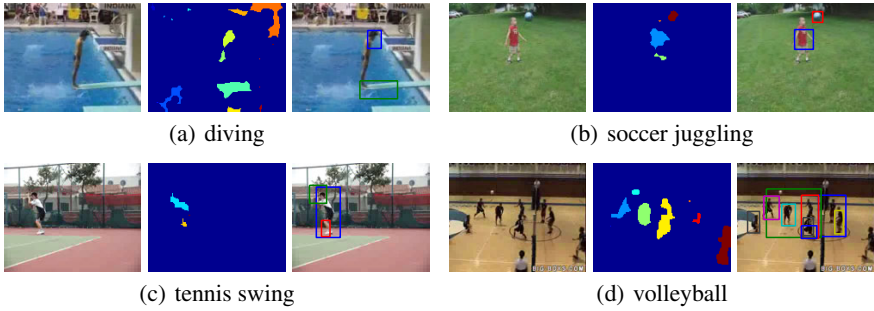
Actions may involve certain objects. For example, for a throwing action, the presence of a ball and/or the shape of its trajectory are important discriminative cues. With this intuition, we find candidate object regions and extract object-centric features from these regions. We do not use any explicit object detector for this purpose. We consider any moving region that has sufficient temporal and spatial coherence as a “*candidate object*”. Once a candidate object region is detected, we try to find the associated tracks and the corresponding features from each track.

**Extracting candidate object tracks:** We extract candidate object regions as follows: First, we estimate the foreground motion as described in Sec 3.1. Then, we find the consistent regions among the estimated foreground motion fields. We do this by looking at temporal, spatial and appearance consistency of the detections in sequential frames. More formally, given a video frame and its estimated foreground flow, we first find the connected components of the flow field, yielding possible object regions. We follow an agglomerative clustering approach to group each of these regions based on their appearance and spatial coherence within the video. In this agglomerative clustering, the similarity between regions is computed by using the  $\chi^2$  distance of color histograms and Euclidean distance of their midpoint coordinates. During this clustering, we allow a temporal gap of up to 10 frames. In the end, the small clusters (i.e., less than 5 frames) are considered to be noise and discarded. This clustering serves as an initial preprocessing step to remove noisy and discontinuous detections.

After this initial step, we form tracks for each remaining region. We follow a greedy approach for generating tracks: we start from the region with the largest area and we track that region using mean-shift tracking [6], forward and backward in time. For the remaining regions, we check if there is already a track that overlaps with that region to a certain degree (30%). The degree of overlap is calculated as the ratio of the intersection over the union of the corresponding bounding boxes. If there is no previous overlapping track, we create a new track. Example outputs of this procedure are shown in Fig. 3.

**Object-centric motion features:** Once the candidate object tracks are found, we extract the motion features from each track. We describe the flow region in each detection bounding box by dividing it into  $2 \times 2$  spatial partitions, and representing each spatial partition with histograms of the four major flow orientations. We extract these motion histograms over a snippet of 5 frames from each track. We apply a windowing scheme





**Fig. 3.** Objects are extracted by grouping the optical flow regions and tracking them over time. Short tracks are eliminated, so that we are left with object regions that have a considerable amount of motion throughout the sequence.

over each object track to extract all the instances that will be input to the MIL algorithm. The final descriptor has  $4 \times 4 \times 5 = 80$  feature dimensions.

**Object-centric shape features:** This feature channel includes shape information for the moving objects in the sequence. With this feature, we aim to capture the immediate context information for the action by defining the shape of nearby objects, such as bicycles, horses or smaller objects, like rackets, balls, etc. We define the shape of object regions by using HOGs. Since the size and scale of the object bounding boxes are not constant and we cannot define a single width/height ratio, we extract HOGs from the spatial grids. That is, instead of using a regular cell and block size, we assume that the object region is divided into an equal number of spatial blocks. We use  $3 \times 3$  spatial bins and represent each spatial bin with nine gradient orientations. We then normalize each descriptor with respect to the object size. The final descriptor size for each object bounding box has  $9 \times 9 = 81$  feature dimensions.

### 3.4 Scene Features

Apart from the person and object related features, the overall properties of the scene can give us related contextual information about the action taking place. For example, if we see a basketball hoop and a court, the probability of observing people playing basketball is higher. In order to exploit these properties, we extract shape and color features.

**Scene shape features:** To describe the scene structure, we extracted Gist [28] features from five frames selected randomly from each video. We use the original parameter settings provided in [28] and the final Gist descriptor has 512 feature dimensions.

**Scene color features:** Color features can be complementary to the shape information for the scene. For example, the presence of a “blue rectangular region” (i.e., a swimming pool) may be helpful in identifying the “diving” action. We extract color features respecting the coarser spatial layout of the scene. For this purpose, we divide each scene horizontally into three equal regions and extract color histograms from each region. For the color histogram, we discretize the RGB colorspace into 16 bins. The final descriptor has  $16 \times 3 = 48$  dimensions. We do this for three randomly selected frames.

## 4 Combining Features - A Multiple MIL Approach

In our problem, we are given a set of videos with labels that tell us the presence of an action class in each video. However, we do not know the exact spatio-temporal location of the specified action in each video, nor do we know what related objects or scene information will contribute to the identification of that class. There may be many object and/or person tracks extracted from each video. Some of these tracks may be relevant to the action, e.g., the track of a basketball or a jumping person, whereas some of the tracks may be irrelevant or caused by noise.

This scenario suggests the particular suitability of “multiple instance learning” (MIL) [2]. In MIL, the given class label is associated with bags (rather than instances as in the case of fully supervised learning), where each bag consists of one or more instances. A bag is labelled as positive if at least one instance  $x_{ij}$  in the bag is known to be positive. A bag is labelled as negative if all the instances in that bag are known to be negative. Individual labels of the instances are unknown. Since the labels are given to bags rather than instances, the learning procedure operates over the bags.

In our case, a bag contains all the instances extracted from a video sequence for a particular feature channel. For example, for the Gist feature, the bag would contain five instances, one Gist feature vector per each of the randomly selected frames from the video. For the person-centric motion feature, there would be several feature vectors  $x_{ij}$  extracted by employing the windowing procedure over each detected person track and each of these feature vectors is considered to be an instance inside the bag of the corresponding feature channel.

Formally, for each video  $i$ , we have one bag  $\mathbf{B}_i^f$  per feature channel  $f \in \{1, \dots, F\}$ . Each  $\mathbf{B}_i^f$  contains multiple instances  $x_{ij}$  such that  $\mathbf{B}_i^f = \{x_{ij}^f : j = \{1, \dots, n_i^f\}\}$ . Here,  $n_i^f$  is the number of instances of that feature type in video  $i$ . Each bag has an associated label  $Y_i \in A$ , where  $A = \{a_1, \dots, a_M\}$  is the possible set of  $M$  actions.

In order to represent these bags in the MIL framework, we first embed the original feature space  $x$ , to the instance domain  $\mathbf{m}(B)$ , via the instance embedding framework of [5]. In [5], each bag is represented by its similarity to each of the instances in the dataset. In our case, this is infeasible, given the large size of the dataset and number of instances per bag. Therefore, we cluster the data using k-means to find potential target concept instances  $c_l^f \in C^f$ . We do this for each action class separately, setting  $k$  to a constant value (we use  $k = 50$ ). The total size of  $C^f$  becomes  $N = k \times M$  for each feature channel. The similarity between bag  $\mathbf{B}_i$  and concept  $c_l^f$  is defined as

$$s(c_l^f, \mathbf{B}_i^f) = \max_j \exp \left( -\frac{D(x_{ij}, c_l^f)}{\sigma} \right), \quad (2)$$

where  $D(x_{ij}, c_l^f)$  measures the distance between a concept instance  $c_l^f$  and a bag instance  $x_{ij}$ . In our case, since all the features are histogram-based, we can use the  $\chi^2$  distance  $D(x_{ij}, c_l^f) = \chi^2(x_{ij}, c_l^f) = \frac{1}{2} \sum_d \frac{(x_{ij}(d) - c_l^f(d))^2}{x_{ij}(d) + c_l^f(d)}$ , where  $d$  is a feature dimension of the instance feature vector. For the bandwidth parameter  $\sigma$ , we use the standard deviation of each feature embedding.

Each bag can then be represented in terms of its similarities to each of these target concepts and this mapped representation  $\mathbf{m}(B_i^f)$  can be written as

$$\mathbf{m}(B_i^f) = [s(c_1^f, B_i), s(c_2^f, B_i), \dots, s(c_N^f, B_i)]^T. \tag{3}$$

We convert the instances from each feature channel to their MIL representation separately. Subsequently, we need a way to combine these different feature channels. For this purpose, we propose two combination techniques. The first technique concatenates all feature channels and treats the problem as a classification problem over the joint set of features. More formally, in our first method, we represent each bag  $B_i$  with its concatenated embeddings over  $F$  feature channels, such that

$$\hat{\mathbf{m}}(B_i) = [\mathbf{m}(B_i^1)\mathbf{m}(B_i^2) \dots \mathbf{m}(B_i^F)]. \tag{4}$$

We use an L2-regularized linear SVM for the classification over these concatenated bag representations  $\hat{\mathbf{m}}(B)$ . In this way, the positivity constraint of the MIL framework is extended over multiple feature channels. If a  $B_i^f$  is empty for a particular  $f$ , we simply assign the corresponding  $\mathbf{m}(B_i^f)$  to zero.

In the above formulation, each feature channel is treated equally. However, there may be certain cases where a particular feature channel is more informative than the other feature channels for a specific action. Likewise, some of the feature channels may contain redundant information for specific actions. In this case, we may be interested in learning global weights for individual feature channels.

This observation motivates the formulation of our second method, which employs a joint formulation for learning the global weights for feature channels. This global weighting is analogous to learning the kernel weights in multiple kernel learning (MKL) [11]. In MKL, the task is to select informative kernels, whereas here we try to select informative feature channels. We formulate the optimization as follows:

$$\begin{aligned} \min_{w, \alpha, b} \quad & \sum_f (w^f)^T w^f + \beta \alpha^T \alpha + \gamma \sum_i L \left( y_i, \sum_f \alpha_f (w^f)^T m^f + b \right), \tag{5} \\ \text{s.t.} \quad & \alpha \geq 0 \end{aligned}$$

In this formulation  $w^f$  is the weight vector for individual features in  $f$ th feature channel,  $\alpha_f$  is the global weight of the whole feature channel,  $L$  is the loss function (we use Hinge loss).  $\beta$  and  $\gamma$  are the regularization parameters and  $b$  is the bias term. Here, each  $\alpha_f$  defines a global combination weight for the instances of feature channel  $f$ . The first term in this objective function in Eq. 5 stands for the regularization of the individual feature weights on each channel, whereas the second term corresponds to the regularization on global feature channel weights. If a feature channel tends to be very noisy, this global weighting scheme can help in deemphasizing that feature channel completely by assigning the corresponding  $\alpha_f$  to a small value.

The objective function in Eq. 5 becomes convex when the  $\alpha$  or  $w$  vector is fixed. Therefore, we follow an iterative alternating optimization approach in the primal space, which is a coordinate descent method. In this iterative approach, we first fix  $\alpha$  and solve for  $w$  and  $b$ , such that

$$\min_{w,b} \sum_f (w^f)^T w^f + C \sum_i L \left( y_i, \sum_f \sum_{z \in G_f} (w_z^f)^T (\alpha_f m_z^f) + b \right). \quad (6)$$

Here,  $G_f$  represents the group of features for feature type  $f$ . Once Eq. 6 is optimized with respect to  $w$  and  $b$ , we then fix the  $w$  and  $b$ , and optimize  $\alpha$  such that,

$$\min_{\alpha} \beta \alpha^T \alpha + C \sum_i L \left( y_i, \sum_f \alpha_f ((w^f)^T m^f) + b \right). \quad (7)$$

Note that in this formulation, both steps minimize the same objective, so convergence is guaranteed. In our experiments, we observe that convergence to a local minimum is achieved in  $\approx 10$  iterations.

## 5 Experiments

In order to test our approach, we use the YouTube dataset collected by Liu et al [22]. This is a very large dataset that consists of 1168 videos in total. This is a particularly suitable dataset for studying the effects of object and scene properties of actions, since there are actions involving specific objects (like basketball) and scenes (like diving). The dataset contains videos of 11 actions; these are basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. It is a quite challenging dataset with lots of camera movement, cluttered backgrounds, different viewing directions and varying illumination conditions. Videos for each category of action are divided into 25 related subsets, and leave-one-out cross validation is applied over these subsets, following the same evaluation methodology of [22].

### 5.1 Evaluation

Table 1 summarizes the overall quantitative results. The classification results here are normalized with respect to the number of videos for each action type.

We first evaluate the performance of the individual feature channels. The first six rows of Table 1 include the individual classification accuracies for each of the feature channels represented in the embedded MIL domain. Note that, in our setup, the object tracks are allowed to overlap with or include person tracks, so the object tracks may sometimes include person track information as well. The results show that, even using the single feature channel Gist gives 53.20% average classification accuracy in this dataset, whereas the simple color histogram feature is able to perform with 49.28% average accuracy. These numbers are noticeably high, compared to the chance level in this dataset, which is 9.09%. This observation shows that using scene features can provide a great deal of useful information about the possible action, especially in this dataset. For example, for the diving action, the simple color features achieve 86% accuracy, whereas for volleyball spiking, the Gist features give 81%. These results suggest that when the person is less visible, the scene features can be used for reducing the set of possible actions considered. On the other hand, where the person is more visible (e.g. videos of

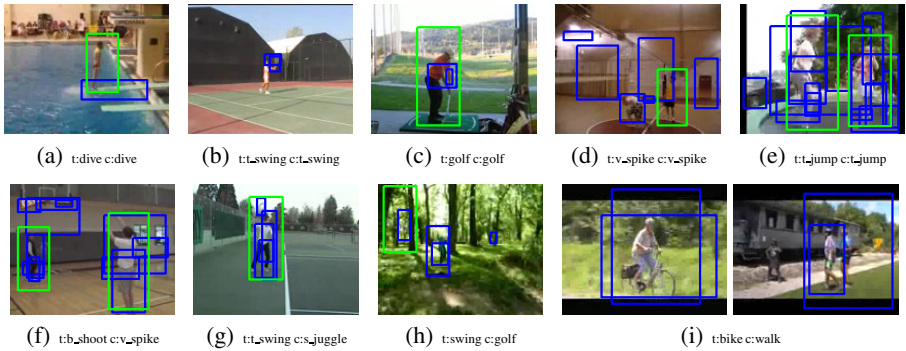
**Table 1.** Overall performance evaluation of individual feature channels and their combination. perOF/objOF: person/object optical flow features, perHOG/objHOG: person/object HOG features. p+s: person and scene features, p+o : person and object features and o+s: object and scene feature combinations. The best results for each action class are shown in bold. We see that action recognition benefits from both object and scene features in most of the action types.

% correct classification using single feature channels												
	b_shoot	bike	dive	golf	h_ride	s_juggle	swing	t_swing	t_jump	v_spike	walk	Avg
perOF	20.20	44.83	51.0	69.0	45.0	44.0	36.0	32.0	64.0	29.0	29.27	42.72
perHOG	28.28	57.93	56.0	40.0	51.0	36.0	43.0	45.0	34.0	49.0	39.84	43.64
objOF	14.14	45.52	24.0	36.0	51.0	20.0	42.0	14.0	59.0	25.0	33.33	33.09
objHOG	21.21	44.14	62.0	55.0	38.0	22.0	42.0	44.0	42.0	45.0	21.95	39.75
gist	38.38	60.69	69.0	61.0	66.0	9.0	42.0	61.0	54.0	81.0	43.09	53.20
color	33.33	44.83	86.0	65.0	43.0	22.0	27.0	47.0	57.0	73.0	43.90	49.28
% correct classification using combinations of channels												
p+s	44.44	70.34	92.0	87.0	63.0	35.0	56.0	75.0	84.0	84.0	56.91	67.97
p+o	40.40	70.34	84.0	91.0	63.0	<b>54.0</b>	63.0	60.0	84.0	78.0	50.41	67.11
o+s	47.47	73.79	91.0	90.0	73.0	35.0	64.0	75.0	83.0	<b>89.0</b>	56.10	70.67
% correct classification using all feature channels												
p+o+s	48.48	<b>75.17</b>	95.0	<b>95.0</b>	<b>73.0</b>	53.0	<b>66.0</b>	77.0	<b>93.0</b>	85.0	66.67	<b>75.21</b>
w[p+o+s]	43.43	<b>75.17</b>	<b>96.0</b>	94.0	72.0	47.0	65.0	74.0	<b>93.0</b>	85.0	67.48	73.83
Liu [22]	<b>53.0</b>	73.0	81.0	86.0	72.0	<b>54.0</b>	57.0	<b>80.0</b>	79.0	73.3	<b>75.0</b>	71.2

trampoline jumping, golf, juggling actions), the optical flow of the person detections seems to be the most informative feature.

Second, we look at the joint performance of these feature channels. In Table 1, “p+o” refers to combination of person-centric and object-centric features together, i.e. the first four feature channels, ignoring the scene dimension. The rows “p+s” and “o+s” correspond to combination of the “person and scene” and the “object and scene” feature channels, respectively. Looking at these feature combinations, we see that, in most of the cases, using them in combination improves classification accuracy significantly. For example, for the trampoline jumping action, the maximum response from the individual feature channels is 64% for person optical flow features, whereas, accuracy increases to 84% if person features are considered in combination with object or scene features.

Third, we look at the overall combination results. As described in Sec 4, we have combined all feature channels using two different methods; the first method uses L2-regularized linear SVM over the concatenated embedded feature channels (represented as p+o+s in Table 1), the second method learns global combination weights over each feature bag (w[p+o+s]). In 9 out of 11 actions, using all the features together yields higher classification accuracy. These results demonstrate that all feature channels are informative and complementary to each other, and that each of them introduces some amount of useful information for the identification of the actions in this dataset. We see that both of the proposed combination techniques introduce an improvement over the best reported results in this dataset [22], while the average improvement is higher without the global weights (4% and 2.6% respectively). We believe that this difference is due to the the high amount of noise in some of the feature channels. The excessive noise may cause the weighting scheme to underestimate the exact weights of that feature channel during training.



**Fig. 4.** Example classification results. The person regions are shown in green and the candidate objects are shown in blue. The subcaptions shows the true class label and output classification label, respectively. See text for details.

b_shoot	0.48	0.12	0.03	0.01	0.05	0.01	0	0.07	0.01	0.14	0.07
bike	0.03	0.75	0.01	0	0.06	0	0.01	0.01	0.01	0.01	0.12
dive	0.03	0	0.95	0	0.01	0.01	0	0	0	0	0
golf	0.02	0.01	0	0.95	0	0	0	0	0	0.01	0.01
h_ride	0.03	0.05	0	0.01	0.73	0.01	0.02	0	0.01	0.02	0.12
s_juggle	0.05	0.06	0.02	0.02	0	0.53	0.07	0.04	0.06	0.04	0.11
swing	0	0.04	0	0.06	0	0.02	0.66	0.01	0.1	0.04	0.07
t_swing	0.06	0.01	0	0.07	0.02	0.05	0	0.77	0	0	0.02
t_jump	0.01	0.03	0	0.01	0	0	0.02	0	0.93	0	0
v_spike	0.07	0	0	0.01	0.01	0	0.02	0	0.01	0.85	0.03
walk	0.01	0.15	0.02	0.02	0.07	0.05	0.01	0.02	0	0	0.67
b_shoot											
bike											
dive											
golf											
h_ride											
s_juggle											
swing											
t_swing											
t_jump											
v_spike											
walk											

**Fig. 5.** Overall confusion matrix of [p+o+s]. The average accuracy is 75.21%.

Example classification results are shown in Fig. 4. The first row shows the correct classifications. For the diving action in Fig. 4(a), both the person and the related object (diving board) are detected and their features are complementary to the scene features. In Fig. 4(b), the person detection has failed, but the tennis racket is found and helps the identification of the tennis swing action. In Fig. 4(e), the candidate objects are noisy, but the person tracks seem reliable. Example misclassifications are shown in the second row of Fig. 4. The failure cases are mostly caused by the multiple people, noisy detections, and/or multiple actions. For instance, in Fig. 4(f) there are multiple people and many candidate objects. In Fig. 4(i), although there is a biking person in the first half of the video, in the second part there is a walking detection.

Figure 5 shows the confusion matrix of our approach. Most of the confusion occurs between walking and biking actions. This is mostly due to the higher frequency of close-up recording in these actions. When there is extensive close-up in the video, the motion stabilization procedure fails to estimate the homography of the scene correctly, because of the increased ratio of the moving regions. Basketball shooting and volleyball

actions are also confused in some cases; this is largely because most of the time, the basketball and volleyball sports use very similar courts.

In a typical video, our moving region grouping procedure results in 20-30 object tracks on average in the YouTube dataset. While the complete annotation of these tracks is infeasible, we estimate that approximately five tracks on average are relevant in each video. The experiments indicate that our framework can succeed, even under such challenging conditions.

## 6 Conclusion

In this paper, we present an approach for combining features of the people, objects and scene for better recognition of actions. The videos available for training our approach are only weakly annotated; we do not know where or when in the video the action occurs, nor do we know which objects or scene features will contribute to the identification of that action. To discover these automatically during training, we use a MIL-based framework.

Our results show that, scene and object properties can indeed be used as complementary to person features for the correct identification of actions. This is especially true when the person is seen from a far distance and the distinct features of the human body are not fully visible. In that case, the moving regions nearby or the overall scene gist can give an idea about what the person/people is up to in that scene.

Action recognition in YouTube videos is an especially good application domain for our method. The low resolution and the unstable camera conditions can make a single feature channel unreliable on its own. In that case, the recognition of actions is likely to benefit from multiple feature channels, as we demonstrate in this work.

We use three main types of features and ignore the temporal and spatial relationships of these features. Our framework can be extended to handle more feature channels, like space-time interest points [29] and can benefit from more complex video object segmentation methods like [15]. Future work includes exploring these techniques and more feature channels, together with their spatio-temporal relationships.

**Acknowledgments.** This material is based upon work supported in part by the U.S. National Science Foundation under Grant No. 0713168.

## References

1. Ali, S., Shah, M.: Human action recognition in videos using kinematic features and multiple instance learning. *IEEE TPAMI* 32(2) (2010)
2. Andrews, S., Tsochantaris, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: *NIPS*, pp. 561–568. MIT Press, Cambridge (2003)
3. Babenko, B., Yang, M.-H., Belongie, S.: Visual Tracking with Online Multiple Instance Learning. In: *CVPR* (2009)
4. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: *ICCV* (2005)
5. Chen, Y., Bi, J., Wang, J.Z.: Miles: Multiple-instance learning via embedded instance selection. *IEEE TPAMI* 28(12), 1931–1947 (2006)
6. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE TPAMI* 25(5), 564–575 (2003)

7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
8. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV '03, pp. 726–733 (2003)
9. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
10. Forsyth, D.A., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational studies of human motion: part 1, tracking and motion synthesis. *Found. Trends. Comput. Graph. Vis.* 1(2-3), 77–254 (2005)
11. Gehler, P.V., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV (2009)
12. Gupta, A., Davis, L.S.: Objects in action: an approach for combining action understanding and object perception. In: CVPR (2007)
13. Han, D., Bo, L., Sminchisescu, C.: Selection and context for action recognition. In: ICCV (2009)
14. Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., Huang, T.S.: Action detection in complex scenes with spatial and temporal ambiguities. In: ICCV (2009)
15. Huang, Y., Liu, Q., Metaxas, D.N.: Video object segmentation by hypergraph cut. In: CVPR (2009)
16. Ikizler, N., Forsyth, D.: Searching for complex human activities with no visual examples. *IJCV* 80(3) (2008)
17. Ikizler-Cinbis, N., Cinbis, R.G., Sclaroff, S.: Learning actions from the web. In: ICCV (2009)
18. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: ICCV (2007)
19. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV (2007)
20. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
21. Liu, F., Gleicher, M.: Learning color and locality cues for moving object detection and segmentation. In: CVPR (2009)
22. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos “in the wild”. In: CVPR (2009)
23. Maron, O., Ratan, A.L.: Multiple-instance learning for natural scene classification. In: ICML, pp. 341–349 (1998)
24. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
25. Mikolajczyk, K., Uemura, H.: Action recognition with motion-appearance vocabulary forest. In: CVPR (2008)
26. Moore, D.J., Essa, I., Hayes, M.H.: Exploiting human actions and object context for recognition tasks. In: ICCV (1999)
27. Niebles, J.C., Han, B., Ferencz, A., Fei-Fei, L.: Extracting moving people from internet videos. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 527–540. Springer, Heidelberg (2008)
28. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV* 42(3), 142–175 (2001)
29. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR (2004)
30. Tran, D., Sorokin, A.: Human activity recognition with metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 548–561. Springer, Heidelberg (2008)
31. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In: CVPR (2008)



# Representing Pairwise Spatial and Temporal Relations for Action Recognition

Pyry Matikainen<sup>1</sup>, Martial Hebert<sup>1</sup>, and Rahul Sukthankar<sup>2,1</sup>

<sup>1</sup> The Robotics Institute, Carnegie Mellon University

<sup>2</sup> Intel Labs Pittsburgh

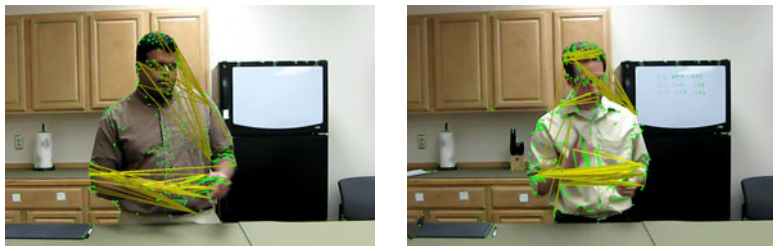
{pmatikai,hebert,rahuls}@cs.cmu.edu

**Abstract.** The popular bag-of-words paradigm for action recognition tasks is based on building histograms of quantized features, typically at the cost of discarding all information about relationships between them. However, although the beneficial nature of including these relationships seems obvious, in practice finding good representations for feature relationships in video is difficult. We propose a simple and computationally efficient method for expressing pairwise relationships between quantized features that combines the power of discriminative representations with key aspects of Naïve Bayes. We demonstrate how our technique can augment both appearance- and motion-based features, and that it significantly improves performance on both types of features.

## 1 Introduction

It is well known that classification and recognition problems in general cannot be solved by using a single type of feature and that, instead, progress lies in combinations of feature representations. But as there is as vast an array of ways to combine and augment features as there are features themselves, the development of such higher order methods is as difficult (and potentially rewarding) an endeavor as direct feature construction. These meta-methods span a range of approaches from those that make absolutely no assumptions on their base features, and thus are consigned to black-boxes operating on vectors of numbers, to those that are so intimately tied to their base features as to be virtually inseparable from them. The former types of methods, such as multiple kernel learning (MKL) techniques are attractive for their broad applicability, but the latter tend to be more powerful in specific applications due to their strong coupling with their underlying features.

However, between those extremes there are still augmentations that compromise between generality and power by being applicable to broad classes of loosely-related base features. The popularity of statistical bag-of-words style techniques [1,2] for action recognition and related video tasks creates an opportunity to take advantage of the broad similarities in these techniques. In particular, these techniques rely on accumulating histograms of quantized features extracted from video, features that are almost always localized in space and time. Yet these methods typically do not take advantage of the spatial



**Fig. 1.** Pairs of features probabilistically vote for action classes; pairs voting for the correct action class are shown in yellow, with brighter color denoting stronger (more informative) votes. For “answerPhone”, the relative motion of the hands is particularly discriminative. These results employ trajectory fragments on the Rochester dataset [3], but our method works with any localized video feature.

and temporal relationships between features. While it is obvious that in general using such relationships should help, the subtleties of designing appropriate representations have limited their use. Figure 1 shows examples of the types of informative relationships between features that we are interested in.

Pairwise spatial relationships, in the form of star topologies, fans, constellations, and parts models, have seen frequent use in static image analysis [4,5,6]. Practical limitations have made transitioning these methods to video difficult. Sparse pairwise topologies (stars, fans, parts) often suffer from a lack of appropriately annotated training data, as they often require annotations that specify the topology for training [7,8]. Alternatively, there are structured methods which can operate without such annotations, but at the cost of significantly more complicated and computationally expensive training or testing [9,10]. In the special limited case of a fixed camera, the entire topology can be fixed relative to the frame by simply using the absolute positions of features [11,12].

The key contributions of this paper can be summarized as follows: (1) we propose an efficient method for augmenting quantized local features with relative spatial-temporal relationships between pairs of features, and (2) we show that our representation can be applied to a variety of base features and results in improved recognition accuracy on several standard video datasets.

For the pairwise model, the most direct representation that is compatible with bag-of-words techniques is to simply generate higher-order features by quantizing the possible spatio-temporal relationships between features. However, this results in a significantly larger number of possible codewords; for example, 100 codeword labels and 10 possible relationships, would produce 100,000 possible labels for the pairwise codewords. Attempts to mitigate this have centered on dimensionality reduction and limiting the number of relationships. In the former case, Gilbert *et al.* [13,14] employ data mining techniques to find frequently-occurring combinations of features. Similarly, Ryoo and Aggarwal [15] use a sparse representation of the resulting high-dimensional histograms, in addition to using a relatively small relationship set. Taken to the extreme, Savarese

*et al.* [16] consider effectively only one relationship: whether two features occur within a fixed distance of each other, and likewise Sun *et al.* [17] also use a simple proximity relationship.

The subtle difficulty is exposing enough information for discriminative machinery to gain traction, but not so much as to overwhelm it in the noise. We strike this balance by selectively organizing estimated pairwise relationships, thereby exploiting fine spatio-temporal relationships without having to resort to an unmanageable representation for the classifier. Inspired by recent work on the max-margin Hough transform [18], we propose accumulating probabilities in a Naïve-Bayes like fashion into a reduced number of bins, and then presenting these binned probabilities to discriminative machinery. By choosing our bins to coincide with codeword labels, we produce vectors with size proportional to the number of codewords while still taking advantage of discriminative techniques.

Since we propose a method for augmenting features with spatio-temporal relationships, we wish to show that this augmentation performs well on a range of features. To this end, we consider two radically different types of base features. First, we consider features built from space-time interest points (STIPs) with associated Histogram of Oriented Gradient (HOG) descriptors, which are sophisticated appearance-based features popularized by Laptev *et al.* [1]. Second, we consider a simple form of trajectory based features similar to those proposed by Matikainen *et al.* [19] and Messing *et al.* [3], quantized through a fixed (training data independent) quantization method. This selection of base features demonstrates the effectiveness of our method on both appearance- and motion-based features, as well as on sophisticated and simple feature extraction methods. In particular, our simplified trajectory method produces a fixed number of features per frame, and the feature labels are not derived from a clustering of training data. The method is virtually certain to produce a large number of extraneous features, and the feature labels are likely to be more sensitive to noise compared to those produced through clustering. In contrast, STIP-HOG produces relatively few features, which tend to be more stable due to the clustering.

As discussed above, our proposed approach formulates the problem in a Naïve Bayes manner, but rather than independently summing per-feature probabilities in log space, we pass them through a discriminative classifier. We train this classifier by estimating all of the cross probabilities for feature labels, that is, for each pair of labels and each action we build a relative location probability table (RLPT) of the observed spatial and temporal relationships between features of those labels under the given action. Then, any feature label can compute its estimate of the distribution over action probabilities using the trained cross-probability maps. These estimates are combined for each feature label, and the final feature vector is presented to a classifier.

## 2 Base Features

The proposed method can augment a variety of common features employed in video action recognition. To demonstrate our method’s generality, we describe

how it can be applied to two types of features that represent video in very different ways, as discussed below.

## 2.1 Base Feature: STIP-HOG

Laptev *et al.*'s space-time interest points (STIPs) [1], in conjunction with Histogram of Oriented Gradient (HOG) descriptors have achieved state-of-the-art performance on a variety of video classification and retrieval tasks. A variable number of STIPs are discovered in a single video and the local space-time volume near each interest point is represented using an 72-dimensional descriptor. These HOG descriptors are quantized using a codebook (typically pre-generated using k-means clustering on a large collection) to produce a discrete label and a space-time location  $(x, y, t)$  for each STIP.

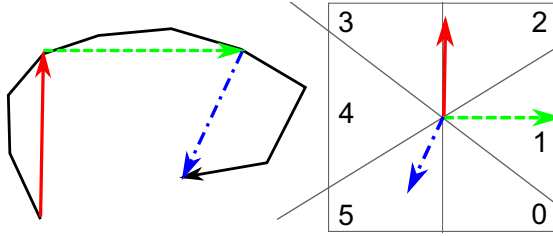
## 2.2 Base Feature: Quantized Trajectories

In previous work [19], we considered trajectory-based features that we coined *trajectons*; these features describe video data in a very different manner from STIP-HOG. First, Harris corner features are tracked in a given video using KLT to produce a set of trajectories. Each trajectory is first converted from a list of  $(x_t, y_t)$  position pairs into a list of discrete derivative pairs  $(dx_t, dy_t) = (x_t - x_{t-1}, y_t - y_{t-1})$ . These trajectories are then broken up into overlapping windows of fixed duration  $T$ , each of which is considered a new feature or trajectory fragment. Unlike STIP-HOG, trajectory fragments seek to express longer-term motion in the video. We generally follow our previous work, but substitute a more straightforward quantization method.

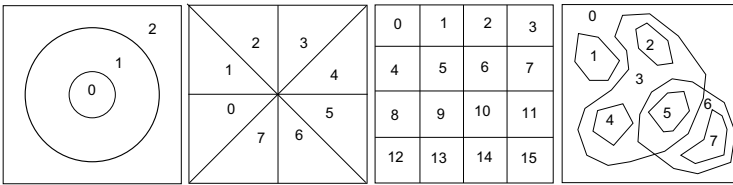
**Sequencing code map (SCM) quantization.** In our earlier work, we quantize trajectories using k-means. Fragments are clustered to produce a codebook. Messing *et al.* [3] also use trajectory features but employ a different quantization strategy in which trajectories are soft-assigned to Markov mixture components; their strategy is similar to that of Sun *et al.* [17] who also consider quantized transitions within a trajectory.

Both Messing *et al.*'s and our earlier approach can be computationally expensive depending on the number of mixture components or k-means centers, respectively. Taking inspiration from both, we propose quantizing fixed-length trajectories using a derivative table similar to Messing *et al.*'s, which we call a sequencing code map (SCM), examples of which can be seen in Figure 3. However, rather than using quantized derivatives to look up probabilities, we simply combine the quantized indices over the fixed length trajectory fragment into a single label encoding quantized derivatives at specific times with each fragment (see Figure 2).

In our method, a trajectory fragment is divided into  $k$  consecutive stages of length  $t$  frames, such that  $kt \leq T$ , where  $T$  is the total length of the fragment. The total motion, or summed derivative, of each stage is computed as a  $(dx, dy)_k$  pair for each stage. This  $(dx, dy)$  vector is quantized according to the SCM into one



**Fig. 2.** Sequencing code map (SCM) quantization breaks a trajectory fragment into a number of stages (in this case three) that are separately quantized according to a map (in this case a 6-way angular map). These per-stage labels, called sequence codes, are combined into a final label for the fragment, which in this case would be  $215_6 = 83_{10}$ .



**Fig. 3.** Examples of possible sequencing code maps (SCMs) or relative location maps (RLMs). Our experiments focus on angular maps (second from left) but our method is completely general.

of  $n$  stage labels, or sequence codes; the  $k$  sequence codes are combined to produce a single combined label that can take on  $k^n$  values. For example, with 3 stages and an SCM containing 8 bins, there would be  $8^3$  or 512 labels total. Since the time to quantize a stage is a single table lookup regardless of how the table was produced, this method is extremely computationally efficient (*i.e.*, the computation time does not grow with an increasing number of quantization bins).

Formally, we denote by  $M(dx, dy)$  the SCM function, which is implemented as a two-dimensional lookup table that maps from a  $dx, dy$  to an integer label in the range of 0 to  $n - 1$  inclusive. We denote by  $(dx, dy)_k$  the derivative pair for stage  $k$ . Then the assigned label is given by  $l = \sum_{j=0}^k n^j \cdot M(dx_j, dy_j)$ .

Besides the convenience of not having to build a codeword dictionary and the reduced computational cost, our introduction of this quantization method is meant to demonstrate that our pairwise features do not depend on data-driven clustering techniques. The quantized labels produced by SCM quantization are unlikely to correspond nicely to clusters of features in the dataset (*e.g.*, parts), yet the improvement produced by our pairwise relationships persists.

### 3 Augmenting Features with Pairwise Relationships

In the following subsections we detail our approach for augmenting generic base features with spatial and temporal relationships. While the previous discussions

focused on STIP and trajectory fragment features, our proposed method for pairwise spatio-temporal augmentation applies equally well to any type of feature that can be quantized and localized. In the remainder of this section, a “feature” is simply the tuple  $(l, x, y, t)$ : a codeword label in conjunction with its spatio-temporal position. The set of all observed features is denoted  $F$ .

### 3.1 Pairwise Discrimination with Relative Location Probabilities (RLPs)

Starting with the observation that Naïve Bayes is a linear classifier in log space, in this section we formulate the pairwise representation first in the familiar terms of a Naïve Bayes classifier, and then demonstrate how to expose more of the underlying structure to discriminative methods.

We start with the assumption that all pairs are conditionally independent given the action class. Then, if features are quantized to  $L$  labels and the spatial relationships between features are quantized to  $S$  labels, we could represent the full distribution over pairs of features with a vector of  $L^2S$  bins. Unfortunately, for even as few as  $L = 100$  trajectory labels and  $S = 10$  spatial relationships, there would be  $(100^2)(10) = 100,000$  elements in the computed feature vector, far too many to support with the merely hundreds of training samples typically available in video datasets.

This feature vector must be reduced, but the direct approach of combining bins is just equivalent to using a coarser quantization level. Instead, taking inspiration from the Max-Margin Hough Transform [18] and Naïve Bayes, we build probability maps of the spatial relationships between features, and instead of summing counts, we accumulate probabilities, allowing pairs to contribute more information during their aggregation.

Specifically, we produce a feature vector  $B$  of length  $AL$ , where  $A$  is the number of action classes (*e.g.*, walk, run, jump, *etc.*), and where each entry  $B_{a,l}$  corresponds to the combined conditional probability of all pairs containing a feature with label  $l$  given the action class  $a$ . In other words, a bin contains a feature label’s probabilistic vote for a given action class, and we could compute a Naïve Bayes estimate of the probability of all the observed features given an action by summing all the votes for that action:  $\log P(F|a) = \sum_{l \in L} B_{a,l}$ . However, instead of summing these in a Naïve Bayes fashion, we present the vector as a whole to discriminative machinery, in our case a linear SVM. We now describe how we accomplish this feature vector reduction.

**Notation.** Formally, a video segment has a number of quantized features computed from it. A feature  $f_i \in F$  is associated with a discrete quantized label  $l_i \in L$  as well as a spatio-temporal position  $(x_i, y_i, t_i)$  indicating the frame and location in the frame where it occurs. For a pair of features within the same frame and a given action  $a \in A$ , there is a probability of the two features occurring together in a frame  $P(l_i, l_j|a)$  as well as the relative location probability (RLP) for their particular spatial relationship  $P(x_i, x_j, y_i, y_j|l_i, l_j, a)$ . We make the simplifying assumption that RLPs depend only on the relative spatial relationship between

the two features, so that  $P(x_i, x_j, y_i, y_j | l_i, l_j, a) = P(dx, dy | a, l_i, l_j)$ , where  $dx$  and  $dy$  are slight abuses of notation that should be understood to mean  $x_i - x_j$  and  $y_i - y_j$  where appropriate. This assumption enforces the property that the computed relationships are invariant to simple translations of the feature pairs.

**Probabilistic formulation.** The reduction to a feature vector of length  $AL$  is done by selectively computing parts of the whole Naïve Bayes formulation of the problem. In particular, a full probabilistic formulation would compute  $P(F|a)$  and select the  $a$  that maximizes this expression. Since Naïve Bayes takes the form of multiplying a number of features' probabilities, or in this case pair probabilities, we can exploit the distributive and commutative properties of multiplication to pre-multiply groups of pair probabilities together, and then return those intermediate group probabilities rather than the entire sum. This can be seen as binning the pair probabilities.

Assuming feature pairs are conditionally independent, we can compute the probability of a feature set  $F$  given an action  $a$  according to the equation

$$P(F|a) = \prod_{f_i \in F} P(l_i|a) \prod_{f_j \in F} P(l_j|l_i, a) P(dx, dy|a, l_i, l_j), \quad (1)$$

which strictly speaking double-counts pairs since each pair is included twice in the computation; however, since we are only interested in the most likely action, this is not an issue.

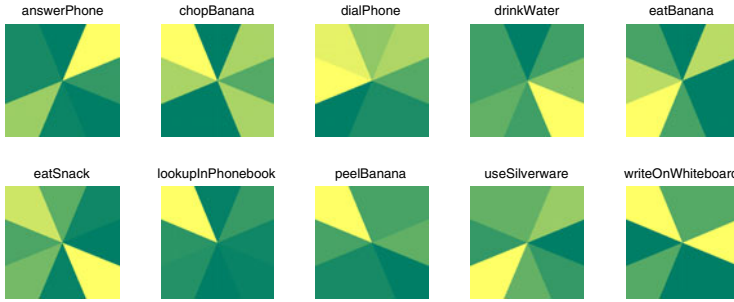
In practice, we employ log probabilities both to avoid issues with numerical precision from extremely small values and to formulate the problem as a linear classifier. In this case the log probability expression becomes

$$\begin{aligned} \log(P(F|a)) &= \sum_{f_i \in F} \log(P(l_i|a)) + \\ &\sum_{f_j \in F} \log(P(l_j|l_i, a)) + \log(P(dx, dy|a, l_i, l_j)). \end{aligned} \quad (2)$$

To simplify the expression we assume uniform probabilities for  $P(l_i|a)$  and  $P(l_j|l_i, a)$ . Later we can include nonuniform label probabilities by simply concatenating the individual label histogram to the pairwise feature vector when both are presented to the classifier. Thus, our probability expression becomes

$$\begin{aligned} \log(P(F|a)) &= \\ &\sum_{f_i \in F} \sum_{f_j \in F} \log(P(dx, dy|a, l_i, l_j)) + C, \end{aligned} \quad (3)$$

which is simply a formal way of stating that the whole log probability is the sum of all the pairwise log probabilities. Since we are only interested in the relative probabilities over action classes, we collect the uniform probabilities for labels into a constant  $C$  which does not depend on the action  $a$ , and which is omitted from the following equations for clarity. We now wish to divide this expression into a number of sub-sums that can be presented to a classifier, and



**Fig. 4.** Relative location probabilities (RLPs) for feature labels 25 and 30 over all actions in the Rochester dataset, using an 8-way angular map. Lighter (yellow) indicates higher probability. We see that for the answerPhone action, features with label 30 tend to occur up and to the right of features with label 25, whereas for useSilverware, features with label 30 tend to occur down and to the left of those with label 25.

this expression leaves us a great deal of flexibility, since we are free to compute and return sub-sums in an arbitrary manner.

**Discriminative Form.** We rewrite Equation 3 in such a way as to bin probabilities according to individual feature labels. In particular, we can rewrite it in log form as

$$\log(P(F|a)) = \sum_{l \in L} \log(P(b_l|a)), \quad (4)$$

where

$$\log(P(b_l|a)) = \sum_{f_i \in l} \sum_{f_j} \log(P(dx, dy|a, l, l_j)). \quad (5)$$

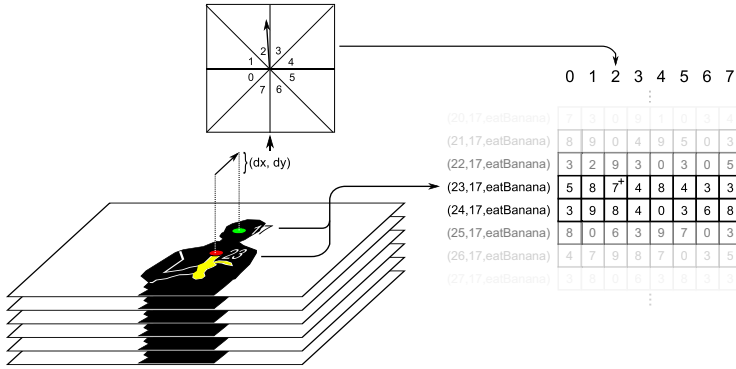
The expression  $\log(P(b_l|a))$  is the bin probability, which directly corresponds to an element of the feature vector according to  $B_{a,l} = \log(P(b_l|a))$ . Since there are  $A$  actions and  $L$  labels, this  $B$  vector contains  $AL$  elements.

### 3.2 Estimating Relative Location Probabilities from Training Data

The previous section assumed that the relative location probabilities (RLPs) were simply available. However, these probabilities must be estimated from real data, requiring some care in the representation choice for the relative location probability tables (RLPTs). An RLPT represents an expression of the form  $\log(P(dx, dy|a, l_i, l_j))$ , where  $a$ ,  $l_i$ , and  $l_j$  are considered fixed. In practice this means that it must represent a function from a  $(dx, dy)$  pair to a log probability, and that we must represent one such function for every  $(a, l_i, l_j)$  triplet of discrete values. While many representations are possible, we use an approach similar to that used for staged quantization.

We denote by  $M(dx, dy)$  a function that maps a  $(dx, dy)$  pair to an integer bin label, allowing  $n$  such labels. We refer to this map as the relative location





**Fig. 5.** Features with labels 17 and 23 are observed in a frame of a training video of the class eatBanana. The feature with label 17 has a relative displacement of  $(dx, dy)$  from that with label 23, which maps to bin #2 in an 8-way angular RLM. Thus, we increment bin #2 in the corresponding table entry; these counts are all converted to estimated log probabilities after the entire video collection is processed.

map (RLM), possible forms of which can be seen in Figure 3. Then the RLPT for a given  $(a, l_i, l_j)$  triplet is a list of  $n$  numbers, denoted  $T_{a,l_i,l_j}$ . An RLP can then be retrieved according to:

$$\log(P(dx, dy|a, l_i, l_j)) = T_{a,l_i,l_j}[M(dx, dy)]. \tag{6}$$

For example, with 216 labels, 10 actions, and 8 bins in the RLM, storing all the RLPs would require  $(216)(10)(8) = 3,732,480$  entries in 466,560 tables.

Estimating the RLPTs is simply a matter of counting the displacements falling within each bin (see Figure 5), and finally normalizing by the total counts in each map. Since some bins may receive zero counts, leading to infinities when the log probability is computed, we use a prior to seed each bin with a fixed number of pseudo-counts. Examples of RLPTs found in this way can be seen in Figure 4.

This method could seemingly generate very sparse probability maps where most bins receive few or no real counts. However, in practice almost all of the bins receive counts. A typical situation (in this case our experiments on Rochester’s Daily Living dataset) might have 10 classes, 216 feature labels, and a probability map with 8 bins, for a total of  $(216^2)(10)(8) = 3.7 \cdot 10^6$  bins. For Rochester we have approximately 120 training videos, each of which is approximately 600 frames long. If we track 300 features per frame, and only consider in-frame pairs, then across all videos we will have  $(300^2)(600)(120) = 6.5 \cdot 10^9$  pairwise features. Thus, on average each bin will receive over a thousand counts.

### 3.3 Extension to Temporal Relationships

The method is naturally extended to include temporal relationships. Rather than representing the relationship between two features as a  $(dx, dy)$  pair, it is

represented as a  $(dx, dy, dt)$  triple. The RLPT then contains entries of the form  $\log(P(dx, dy, dt|a, l_i, l_j))$ , which are indexed according to a mapping function  $M(dx, dy, dt)$ , so that

$$\log(P(dx, dy, dt|a, l_i, l_j)) = T_{a, l_i, l_j}[M(dx, dy, dt)]. \quad (7)$$

Previously, the map  $M$  could be stored as a simple image, whereas with spatial-temporal relationships this map is a volume or series of images. When counts are accumulated or probabilities evaluated, only pairs of features within a sliding temporal window are considered, since considering all pairs of features over the entire video would both result in a prohibitively large number of pairs and prevent the method from being run online on a video stream. Nevertheless, the change from considering only pairs within a frame to pairs within a temporal window vastly increases the number of pairs to consider, and depending on the number of features generated by a particular feature detector, it may be necessary to randomly sample pairs rather than considering them all. We find that for STIP-HOG we can consider all pairs, while for SCM-Traj we must sample.

### 3.4 Classification

We train a linear SVM [20] to classify video clips, which is a straightforward matter of presenting computed  $B$  vectors and corresponding ground truth classes from training clips. Each bin in  $B$  can be interpreted as a particular label’s vote for an action, in which case the classifier learns the importance of each label’s vote.

Since, when considered in isolation, pairwise relationships are unlikely to be as informative as the base features from which they are derived, we present a simple method for combining the raw base feature histograms with the computed pairwise log probability vectors. We do not present this combination method as the canonical way of combining the two sources of information, but rather as a convincing demonstration that the proposed pairwise relationships provide a significant additional source of information rather than merely a rearrangement of the existing data.

Supposing that  $H$  represents the histogram for the base features, and  $B$  represents the computed pairwise relationship vector, then one way of combining the two would be to simply concatenate the two vectors into  $[H, B]$ , and present the result to a linear SVM. However this is unlikely to result in the best performance, since the two vectors represent different quantities. Instead, we separately scale each part, and then simply cross validate to find the scaling ratio  $p$  that maximizes performance on the validation set, where the combined vector is  $[pH, (1 - p)B]$ . This scaled vector is simply presented to the SVM.

## 4 Evaluation

We evaluate our pairwise method on a forced choice action classification task on two standard datasets, the UCF YouTube dataset (UCF-YT) [21] and the

**Table 1.** Action recognition accuracy on standard datasets. Adding pairwise features significantly boosts the accuracy of various base features.

Method	UCF-YT Rochester	
STIP-HOG (single) (Laptev <i>et al.</i> [1])	55.0%	56.7%
STIP-HOG (NB-pairwise alone)	16.4%	20.7%
STIP-HOG (D-pairwise alone)	46.6%	46.0%
STIP-HOG (single + D-pairwise)	59.0%	64.0%
STIP-HOG-Norm (single) (Laptev <i>et al.</i> [1])	42.6%	40.6%
SCM-Traj (single)	42.3%	37.3%
SCM-Traj (NB-pairwise alone)	14.3%	70.0%
SCM-Traj (D-pairwise alone)	40.0%	48.0%
SCM-Traj (single + D-pairwise)	47.1%	50.0%

recently-released University of Rochester Activities of Daily Living [3]. To evaluate the contribution of our method for generating pairwise relationships, we consider two different types of base features: the trajectory based features we introduced earlier, and Laptev *et al.*'s space-time interest points. We consider both our discriminative formulation (denoted D-pairwise) and a Naïve-Bayes formulation (NB-pairwise) for our pairwise features, where the NB-pairwise results are primarily intended as a baseline against which to compare.

Table 1 summarizes our results. Our experiments are designed to evaluate the effect of adding spatial and temporal relations to the features and to understand in detail the effect of various parameters on the performance of the augmented features. Clearly, significantly more tuning and additional steps would go into building a complete, optimized video classification system. In particular, we do not claim that our performance numbers are the best that can be obtained by using complete systems optimized for these data sets. We use the evaluation metric of total accuracy across all classes in an  $n$ -way classification task.

On both datasets we use 216 base feature codewords for both trajectories and STIP-HOG. The number 216 results from the choice of three stages with a 6-way mask for the staged quantization ( $6^3 = 216$ ), and we use the same number for STIP-HOG to make the comparison as even as possible. Likewise, for both datasets we use an 8-way spatial relationship binning for the RLPTs. Combined results are produced by cross validating on the scaling ratio.

UCF-YT consists of 1600 videos in 11 categories acquired from YouTube clips. For evaluation, we randomly split the dataset into a training set of approximately 1200 videos and a testing set of approximately 400 videos. This dataset was chosen for its difficulty, in order to evaluate the performance of pairwise relationships outside of highly controlled environments. In particular, this dataset is challenging because the videos contain occlusions, highly variable viewpoints, significant camera motion, and high amounts of visual clutter.

On UCF-YT we find that discriminative pairwise features are not as informative as the base features, which is not unexpected since the diversity of the dataset means there are unlikely to be strong, consistent relationships between

**Table 2.** Action recognition accuracy with temporal relationships on UCF-YT

Method	STIP-HOG	Traj-SCM
NB-Pairwise (baseline)	16.4%	14.3%
NB-T-Pairwise	22.2%	31.2%
D-Pairwise (baseline)	46.6%	40.0%
D-T-Pairwise	49.2%	39.7%

features. Nevertheless, we still find modest gains for combinations of pairwise and individual features, on the order of 5%. This means that the pairwise features are providing an additional source of information, rather than just obfuscating the information already present in the individual feature histograms. The Naïve Bayes pairwise evaluation performs poorly, but better than chance.

Furthermore, we can see that the performance of our simple fixed quantization on trajectories performs similarly to normalized STIP-HOG features, but significantly worse than non-normalized STIP histograms. This suggests that much of the discriminative power of STIP features might originate from the variation in the quantity of features found in different videos.

The Rochester dataset consists of 150 videos of five individuals performing a series of scripted tasks in a kitchen environment, acquired using a stationary camera. Due to the limited pool of available data, we evaluate using 5-fold cross-validation, using videos from four individuals for training and the fifth for testing, in each fold.

On Rochester we observe that the pairwise features for STIP-HOG do not perform as well as the individual STIP-HOG features, but that the combination outperforms both, which is consistent with the results for UCF-YT. For trajectory features, the pairwise features alone significantly outperforms the base features, a reversal from UCF-YT. The combination of the two outperforms both the individual and pairwise, but adds only a modest gain on top of the pairwise performance.

For both types of features, the gains with pairwise relationships in combination are much larger than for UCF-YT, which is explained by the greater consistency of spatial relationships between codeword labels due to the fixed viewpoint and highly consistent actions. Qualitatively examining which pairs contribute to a correct action identification supports this hypothesis: as can be seen in Figure 1, the pairwise features supporting an action appear to be strongly tied to that action. For STIP-HOG, the Naïve-Bayes pairwise formulation once again performs poorly, however for trajectories the Naïve-Bayes pairwise is the strongest performer. This suggests that for some applications, even simple relationships can give very good performance.

#### 4.1 Effect of Temporal Relationships

The results with using spatial and temporal relationships on UCF-YT are shown in Table 2 in which X-T-Pairwise denotes the classifier (discriminative or

Naïve-Bayes) augmented with temporal relations. For these results, we have used the same 8-way spatial relationship binning combined with a 5-way temporal binning, for a total of 40 bins. The pairwise relationships are evaluated over a 30 frame sliding window. For STIP-HOG, all pairs within the window are considered, but for trajectories we sample 1/20 of the pairs in the interest of tractability. Note that even with this sampling, a four second UCF-YT clip can produce over 100,000,000 pairs when using trajectory features.

The performance of the discriminative pairwise relationships remains virtually unchanged for Traj-SCM, but there is a modest performance boost for STIP-HOG. The Naïve-Bayes versions continue to perform worse than the discriminative ones, however the temporal relationships have a much larger impact on their performance. The difference is especially dramatic for NB-Pairwise vs. NB-T-Pairwise with STIP-HOG, where the temporal relationships have more than doubled the accuracy from 14.3% to 31.2%.

## 4.2 RLPT Sparsity

Earlier we argued that the relative location probability tables should not be sparse based on a simple counting argument. Empirically, we find that for the Rochester dataset 71.6% of the entries receive counts, and that 91.7% of the tables have at least one count in one of the 8 bins. The number of tables containing at least 100 counts is 41.1%, and 15.2% of tables have over 1000 counts. These numbers validate our original claim that the tables are not sparse.

## 5 Conclusion

We present a simple yet powerful method for representing pairwise spatio-temporal relationships between features in the popular bag-of-words framework. Unlike naïvely expanding codewords to include all possible pairs and relationships between features, our method produces an output whose size is proportional to the number of base codewords rather than to its square, which reduces the likelihood of overfitting and is more computationally efficient. We demonstrate that our method can be used to improve action classification performance with dissimilar STIP-HOG (appearance) and trajectory (motion) based features on two different datasets, and that a discriminative formulation of our pairwise features generally outperforms a Naïve-Bayes classification approach. Although our method takes advantage of spatial relationships, it does not require any additional annotation in the training data, making it appropriate for a wide range of datasets and applications.

As we have only considered simple angular maps, there is potentially still considerable power to be extracted from this method through the careful selection of relative location maps. Additionally, we have presented a binning of probabilities based on codeword label, but an interesting question is whether more intelligent data-driven binnings can be found. We plan to explore these questions in future work.

## References

1. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
2. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. IJCV 79 (2008)
3. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV (2009)
4. Carneiro, G., Lowe, D.: Sparse flexible models of local features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 29–43. Springer, Heidelberg (2006)
5. Crandall, D.J., Huttenlocher, D.P.: Weakly supervised learning of part-based spatial models for visual object recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 16–29. Springer, Heidelberg (2006)
6. Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: CVPR (2007)
7. Zhang, Z.M., Hu, Y.Q., Chan, S., Chia, L.T.: Motion context: A new representation for human action recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 817–829. Springer, Heidelberg (2008)
8. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV (2007)
9. Jiang, H., Martin, D.R.: Finding actions using shape flows. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 278–292. Springer, Heidelberg (2008)
10. Junejo, I.N., Dexter, E., Laptev, I., Pérez, P.: Cross-view action recognition from temporal self-similarities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306. Springer, Heidelberg (2008)
11. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. *Image and Vision Computing* 14 (1996)
12. Makris, D., Ellis, T.: Spatial and probabilistic modelling of pedestrian behaviour. In: BMVC (2002)
13. Gilbert, A., Illingworth, J., Bowden, R.: Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 222–233. Springer, Heidelberg (2008)
14. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: ICCV (2009)
15. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV (2009)
16. Savarese, S., DelPozo, A., Niebles, J., Fei-Fei, L.: Spatial-Temporal correlators for unsupervised action classification. In: WMVC (2008)
17. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: CVPR (2009)
18. Maji, S., Malik, J.: Object detection using a max-margin Hough transform. In: CVPR (2009)
19. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: ICCV Workshop on Video-oriented Object and Event Classification (2009)
20. Chang, C.C., Lin, C.J.: LIBSVM – a library for support vector machines (2001)
21. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos “in the wild”. In: CVPR (2009)

# Compact Video Description for Copy Detection with Precise Temporal Alignment

Matthijs Douze<sup>1</sup>, Hervé Jégou<sup>2</sup>, Cordelia Schmid<sup>1</sup>, and Patrick Pérez<sup>3</sup>

<sup>1</sup> INRIA Grenoble, France

<sup>2</sup> INRIA Rennes, France

<sup>3</sup> Technicolor Rennes, France

**Abstract.** This paper introduces a very compact yet discriminative video description, which allows example-based search in a large number of frames corresponding to thousands of hours of video. Our description extracts one descriptor per indexed video frame by aggregating a set of local descriptors. These frame descriptors are encoded using a time-aware hierarchical indexing structure. A modified temporal Hough voting scheme is used to rank the retrieved database videos and estimate segments in them that match the query. If we use a dense temporal description of the videos, matched video segments are localized with excellent precision.

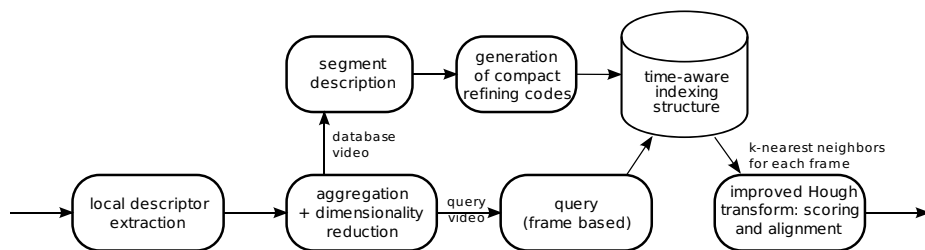
Experimental results on the TRECVID 2008 copy detection task and a set of 38000 videos from YouTube show that our method offers an excellent trade-off between search accuracy, efficiency and memory usage.

## 1 Introduction

We consider the problem of searching a transformed query video, or part of this query, in a large database of videos. This is important, in particular, for detecting video copies that may be illegally delivered on peer-to-peer networks and user generated content sites such as YouTube. The most common transformations observed in practice are camcording and re-encoding, though sophisticated video post-processing is also encountered.

In recent evaluations [1, 2], the use of local descriptors [3–6] combined with a frame voting system appeared to be the most successful architecture for video copy detection. These state-of-the-art systems search individually for each local descriptor of the query video in a structure indexing all local descriptors of the video database. The typical memory requirement associated with representing the set of local descriptors of a video frame ranges from 1 to 10 Kbytes. This seriously limits the number of video frames that can be indexed in practice. Therefore, the video frames are subsampled, which reduces the capability to find very short clips and to determine the precise localization of the query in the database videos. Furthermore, even with subsampling, very large video datasets (several thousands hours of videos) cannot be handled efficiently.

The objective of this paper is to address these scalability and localization issues, while maintaining a high recognition rate. Figure 1 gives an overview of



**Fig. 1.** Overview of our video copy detection method. Local descriptors of a video frame are aggregated into a single vector and the dimension of this vector is reduced. The videos to be indexed are encoded using a temporal-aware indexing scheme. No encoding is applied to the query frame descriptors. A weighted temporal Hough transform provides the final ranking of the database videos w.r.t. the query.

our approach for querying and matching video segments. The individual steps are:

1. Local descriptors are extracted from video frames (either query or database video) and subsequently aggregated into a single vector. This aggregation process is similar to recent approaches for descriptor aggregation [7, 8] which outperform the popular bag-of-features (BOF) representation [9] with descriptors of lower dimension.
2. The dimensionality of this frame descriptor is reduced with either a technique based on a multiplication with a sparse matrix or principal component analysis (PCA).
3. On the database side, the reduced descriptors are encoded within an indexing structure that takes into account the temporal regularity. The video is split in segments. A first description is computed for a segment by minimizing a fidelity criterion for frames of this segment. In the spirit of [10], this segment descriptor is refined by a code based on a product quantizer.
4. Each frame's approximate description is refined by encoding the difference between the frame descriptor and the vector describing the segment it belongs to.
5. A modified temporal Hough voting scheme is used to fuse the votes obtained at the frame level. Its main difference with the conventional method is that the votes are weighted so that their contribution is penalized if 1) the query frame has received a large amount of votes and 2) the database frame has voted several times.

The paper is organized as follows. The frame description method is introduced in Section 2. Section 3 describes how frame descriptors are indexed and retrieved when a query frame descriptor is submitted. The voting scheme is presented in Section 4. The contributions of the different steps are evaluated in Section 5. Furthermore, we compare to the state of the art on the TRECVID 2008 benchmark, and obtain top results in terms of localization accuracy. The scalability of



the approach is demonstrated by experiments on 38000 YouTube videos represented by more than 200 million frames. We show that these videos are indexed in less than 5GB of memory.

## 2 Video Description

### 2.1 Local Description

The first step of our video description extracts a set of local features for each frame. The same approach is used to extract descriptors for the query and database videos. Here, regions are obtained with the scale-invariant Hessian detector [11] and described by CSLBP [12]. Similar to SURF [13] and DAISY [14], this descriptor is a variant of the SIFT descriptor which provides comparable results and reduces the computation time significantly: extracting local descriptors from a frame takes about 35 ms on one 2.4GHz processor core. Note that for large databases, the time for feature extraction is not the critical operation at query time, because it only depends on the query length, not the database size.

### 2.2 Local Descriptor Aggregation: Non Probabilistic Fisher Kernel

Given a set of local descriptors  $\{x_1, \dots, x_i, \dots\}$  for each video frame, it is impossible to store all of them in memory in the case of large scale video search, even if only a few bytes are required for each of them [15, 16]. In general, several hundreds of local descriptors are extracted for each frame.

We, therefore, aggregate the local descriptors into a single vector. We adopt a variant [8] of the Fisher Kernel image representation introduced by Perronnin et al. [7]. The resulting vector, called vector of locally aggregated descriptors (VLAD), provides a compact yet effective representation of images. Assuming that a k-means codebook with  $k$  centroids  $\{c_1, \dots, c_j, \dots, c_k\}$  has been learned, we obtain the VLAD descriptor for a frame, denoted by  $\mu$ , as follows:

1. As for the bag-of-features representation, each local descriptor  $x_i$  of the frame is assigned to the closest centroid in the codebook, i.e., to the quantization index  $\text{NN}(x_i) = \arg \min_j \|x_i - c_j\|$ .
2. Given the set of descriptors assigned to a centroid  $c_j$ , the vector  $\mu^j$  is obtained by summing the differences between these descriptors and the centroid:

$$\mu^j = \sum_{i:\text{NN}(x_i)=j} x_i - c_j. \quad (1)$$

3. The VLAD descriptor associated with a frame is obtained by concatenating the vectors  $\mu^j$  into a single vector.
4. As proposed in [17] for the Fisher Kernel image representation, we apply a power-law normalization to the components to reduce the contribution of the most energetic ones. Here, we use the signed square root of each component. The vector is subsequently L2-normalized and is denoted  $\mu$  in the following.

The resulting vector is of dimension  $k$  times the dimensionality of the input vector, e.g.,  $k \times 128$  for the CSLBP descriptor. For the same codebook size, the dimensionality of the descriptor is significantly larger than for the bag-of-features representation [9]. However, the VLAD description is already discriminant for low values of  $k$  in contrast to BOF, which requires very large codebooks (up to 1 million) to provide the best results [18, 19]. Therefore, the dimensionality of the VLAD vector is typically lower than for BOF. It is worth noting that this representation can be seen as a non-probabilistic version of the Fisher kernel. In the latter, a Gaussian mixture model and soft assignment are used instead of  $k$ -means centroids, and additional information (variance and count) are used to obtain a richer (but longer) representation.

### 2.3 Dimensionality Reduction of Frame Descriptors

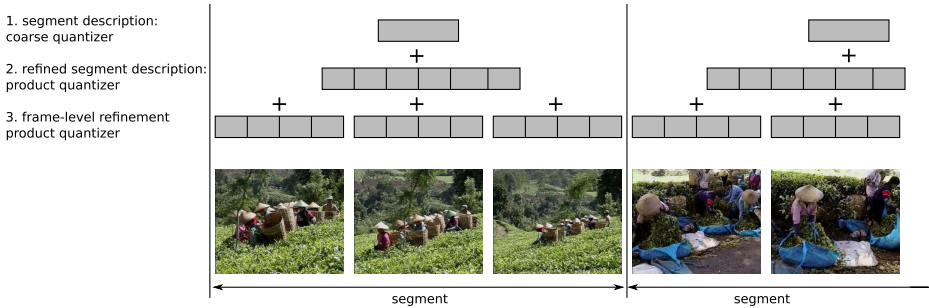
Local descriptor aggregation results in one VLAD descriptor per video frame. This descriptor is highly dimensional: for a typical value of  $k = 64$ , the vector  $\mu$  has  $D = 128 \times k = 8192$  components. Such a vector is difficult to index due to its dimensionality [8]. We, therefore, use and compare two different methods to reduce the dimensionality:

1. Principal component analysis (PCA) allows to reduce the dimension  $D$  of the VLAD descriptor to a smaller dimension  $d$  [8]. The vector  $\mu$  is multiplied with a projection matrix  $M$  formed by the first principal eigenvectors of an empirical covariance matrix. The PCA matrix is pre-multiplied with a random rotation to “whiten” the output;
2. Alternatively, we define  $M$  as a  $d \times D$  sparse matrix obtained as  $M = P\sigma$ , where  $\sigma$  is a  $D \times D$  random permutation matrix and  $P$  is a  $d \times D$  aggregation matrix that sums several consecutive components. For example with  $D = 6$  and  $d = 2$ , a possible matrix  $M$  is:

$$\underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}}_M = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}}_P \times \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_\sigma. \tag{2}$$

The two techniques are evaluated in Section 5. The advantage of using a structured matrix is that there is no need for a training stage. The dimensionality reduction is also cheaper to compute, because the multiplication is more efficient with the sparse matrix  $M$  than with the full matrix obtained by PCA. However, during the search, the dimensionality reduction typically has a low computing cost compared to the other steps.

The descriptor  $f \in \mathbb{R}^d$  of reduced dimensionality is obtained by multiplying the matrix  $M$  with the VLAD descriptor  $\mu$  and by L2-normalizing it. The resulting vector  $f$  is used in the following as the frame descriptor. The dot product is used as a similarity measure.



**Fig. 2.** Hierarchical representation of the frame descriptors. At levels 1 and 2, segments are represented by a frame descriptor. At level 1, the descriptor is quantized to a single index, that is refined at Level 2 by a product quantizer code. At level 3, the individual frames of the segment are represented as a refinement of their segment descriptor. Typically, a segment descriptor is encoded in 4+64 bytes, while each frame is refined by a 16-byte code.

### 3 Indexing Frame Descriptors with Temporal Integration

The objective of this section is to provide a compact representation and an efficient search mechanism for the frames of a database video. Let us consider a video to be indexed<sup>1</sup>, for which we have extracted a set  $f_1, \dots, f_t, \dots, f_T$  of  $d$ -dimensional descriptors using the method introduced in Section 2. The individual descriptors will be approximated, indexed and searched using three successive refinement levels:

1. joint description of a group of contiguous frames: all the frames associated with the same time segment have the same coarse approximation;
2. refinement of this video segment descriptor;
3. refinement of the individual frame descriptors.

Each of these levels provides an improved approximation of an indexed frame descriptor, as illustrated by Figure 2. This procedure is not used for the frames of the query, i.e., their VLAD descriptors are not approximated. Our approach is, to some extent, similar in spirit to the method proposed in [10]. However, a major difference is the integration of the temporal aspect into the indexing scheme.

#### 3.1 Level 1: Coarse Segment Description

Due to the temporal dependency between frames, contiguous frame descriptors of a video shot are similar. We exploit this property by providing a shared description for consecutive frames. Let us define a segment  $\{t_b, \dots, t_e\}$  as an

<sup>1</sup> We treat several videos as a single long video, except that we constrain a segment of frames not to cross video boundaries.

interval of consecutive frames, for which the same level 1+2 approximation of the descriptor is used.

If segments are of fixed size, we have  $t_e = t_b + 1/r - 1$ , where  $r$  is the ratio between the number of segments and the total number of frames in the video. The first approximation is given by a coarse vector quantizer  $q_c(\cdot)$ , for which the codebook  $\mathcal{C}_c = \{c_1, \dots, c_L\}$  is learned using a spherical k-means algorithm. The coarse quantization index  $i_c(t_b : t_e)$  associated with the segment  $\{t_b, \dots, t_e\}$  aims at best representing the set of frame descriptors  $\{f_{t_b}, \dots, f_{t_e}\}$  with respect to the total square error, i.e.,

$$i_c(t_b : t_e) = \arg \min_{i \in \mathcal{C}_c} \sum_{t=t_b:t_e} \|f_t - c_i\|^2, \quad (3)$$

which is equivalent to

$$c_{i_c(t_b:t_e)} = q_c \left( r \sum_{t=t_b:t_e} f_t \right) \quad (4)$$

The vector  $c_{i_c(t_b:t_e)}$  is the level-1 approximation of the frame descriptors in the segment  $\{t_b, \dots, t_e\}$ . When searching the nearest neighbors of a given query descriptor, only the database descriptors associated with the  $k_c$  closest elements in  $\mathcal{C}_c$  are explored.

We have tested both a fixed and adaptive number of frames per segment. Several variants for selecting keyframes have been tested in the adaptive case. Best results were obtained when constructing the segments based on the consistency of the  $k$ -nearest neighbors in  $\mathcal{C}_c$  for the frame descriptors. However, experimental results showed that no variant is better than uniform subsampling. We, therefore, only use segments of fixed size in the following.

### 3.2 Level 2: Segment Descriptor Refinement

The index associated with a given video segment is not precise, as an approximation with a centroid in  $\mathcal{C}_c$  introduces a large quantization error. Therefore, similar to [10], we refine this first approximation by using a product quantizer  $q_f$ , whose codebook<sup>2</sup> is denoted by  $\mathcal{C}_f$ . The total number of centroids implicitly defined by a product quantizer composed of  $m_f$  subquantizers having  $L_f$  centroids each is equal to  $(L_f)^{m_f}$ . This quantizer aims at reducing, over the set of frames associated with a given segment, the average energy of the error vector  $f_t - c_{i_c(t_b:t_e)}$  made by the first approximation  $q_c(f_t) = c_{i_c(t_b:t_e)}$ . The new approximation of a frame descriptor  $f_t$  associated with the segment  $\{t_b, \dots, t_e\}$  is, therefore, of the form

$$f_t \approx c_{i_c(t_b:t_e)} + c'_{i_f(t_b:t_e)}, \quad (5)$$

<sup>2</sup> A product quantizer decomposes the space into a Cartesian product of low dimensional subspaces and quantizes each space separately. As a result, learning codebooks and searching the quantization index have a low complexity even for very large codebooks. The codebook  $\mathcal{C}_f$  has not to be stored explicitly.

where the centroid  $c'_{i_f(t_b:t_e)} \in \mathcal{C}_f$  is obtained by the minimization

$$c'_{i_f(t_b:t_e)} = \arg \min_{c'_i \in \mathcal{C}_f} \sum_{t=t_b:t_e} \|f_t - c_{i_c(t_b:t_e)} - c'_i\|^2. \quad (6)$$

The minimization is efficiently done using the decomposition associated with the product quantizer. Note that this quantizer  $q_f$  is more precise than the coarse quantizer used in the first stage, because the set of centroids  $\mathcal{C}_f$  that is implicitly defined by the product quantizer is large: it is  $2^{8 \times 64}$  for the typical 64-byte codes we use ( $m_f = 64, L_f = 256$ ). The product quantizer decomposition is used to obtain a complexity comparable to that of a quantizer comprising  $L_f$  elements.

### 3.3 Level 3: Refinement of Individual Frame Descriptors

So far, the frames of a segment are described by the same approximation. We now refine the description of each individual frame  $f_t$  by using another refinement product quantizer  $q_r$  induced by  $m_r$  subquantizers with  $L_r$  centroids each. This quantizer encodes the error resulting from the two previous approximations by minimizing the quantization error of  $f_t$ . For a time instant  $t$  such that  $t \in \{t_b, \dots, t_e\}$ , this is done by quantizing the residual error vector  $f_t - c_{i_c(t_b:t_e)} - c'_{i_f(t_b:t_e)}$ . The frame descriptor  $f_t$  is therefore approximated by

$$\hat{f}_t = c_{i_c(t_b:t_e)} + c'_{i_f(t_b:t_e)} + q_r(f_t - c_{i_c(t_b:t_e)} - c'_{i_f(t_b:t_e)}). \quad (7)$$

### 3.4 Search Procedure

Searching a query frame vector  $y$  in a database of frame descriptors  $\mathcal{B} = \{f_1, \dots, f_T\}$  proceeds in a hierarchical manner.

1. The  $k_c$  nearest neighbors of  $y$  in  $\mathcal{C}_c$  identify the segments to be considered: only those associated with one of the selected  $k_c$  indexes are explored.
2. For each vector  $f_i$  in the set of selected lists, the distance approximation

$$l_2(f_i, y) = l_2(f_i - q_c(f_i), y - q_c(f_i)) \approx l_2(q_f(f_i - q_c(f_i)), y - q_c(f_i)) \quad (8)$$

is efficiently obtained from the quantizer indexes  $q_f(f_i - q_c(f_i))$  by exploiting ADC method of [10]. The best segments corresponding to a query vector are found based on the approximation of the square distance of Equation 8. This step returns the set of the  $k_f$  nearest segment descriptors.

3. The query frame descriptor  $y$  is now compared to all the approximated  $\hat{f}_t$  frame descriptors associated with the  $k_f$  segments found in the previous stage. This step returns a set of  $k_r$  nearest frame descriptors.

### 3.5 Complexity

The cost of searching a frame descriptor in a database containing  $T$  frames is expressed in terms of the number  $C_{\text{dist}}$  of regular  $d$ -dimensional vector comparisons and the amount  $C_{\text{mem}}$  of memory scanned in the indexing structure. These

are given by

$$C_{\text{dist}} = L + k_c L_f + \frac{k_f}{r} \quad (9)$$

and

$$C_{\text{mem}} = \alpha \frac{k_c}{L} r T m_f \log_2 L_f + \frac{k_f}{r} m_r \log_2 L_r, \quad (10)$$

where  $\log_2 L_f = \log_2 L_r = 8$  bits = 1 byte in our case. The factor  $\alpha \geq 1$  accounts for the fact that the probability to assign a frame descriptor to an index is not uniform over  $\mathcal{C}_c$ . As observed in [18] in the context of the BOF representation, this increases the expectation of the number of elements that are processed. We measured that  $\alpha \approx 8.4$  with our parameters. Note that our calculation of  $C_{\text{dist}}$  assumes that the level-2 search is optimized by using look-up tables computed on-the-fly.

## 4 Temporal Alignment: Improved Hough Transform

Once each query frame has been matched to  $k_r$  putative frames of the database, the video search matches a sequence of query descriptors to sequence(s) from the database. This sequence matching can be cast in terms of temporal sequence alignment and addressed by dynamic programming techniques, such as dynamic time warping. However, this type of approaches requires a complete frame-to-frame matching cost matrix, which is not feasible at this stage of the detection system. Furthermore, they require a good initialization of the starting and end-point of the matching sequences.

Simplified approaches can be used instead, e.g., partial alignment [20] or classic voting techniques, such as Hough transform or RANSAC. We adopt a temporal Hough transform [6], as it allows the efficient estimation of 1D temporal shifts. Individual frame votes are re-weighted according to suitable normalizations. In particular, re-weighting is used to reduce the harmful influence of temporal "burstiness", i.e., the presence of query frames that match strongly with multiple, possibly unrelated frames in the database. This is similar to the burstiness of local descriptors in images [21].

### 4.1 Hough Transform

As output of the indexing structure, we have a set of matches, each of them represented by a score  $s(\tau, t) > 0$  between the query frame timestamp  $\tau$  and the database timestamp  $t$ . This score is given by the inner product between the query frame descriptor and the approximation of the database descriptor in Equation 7. We set to 0 the scores of frames that are not matched:

$$s(\tau, t) = \begin{cases} \langle y_\tau, \hat{f}_t \rangle & \text{if } t \text{ is retrieved by the index} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

The temporal Hough transform consists in computing a histogram  $h(\delta)$  to accumulate these scores for all  $\delta = t - \tau$  hypotheses. Denoting by  $\mathcal{Y}$  and  $\mathcal{B} =$

$\{1, \dots, T\}$  the sets of time instants on the query side and the database side, respectively, the score is formally obtained as

$$h(\delta) = \sum_{\tau \in \mathcal{Y}} s(\tau, \tau + \delta), \quad (12)$$

where  $s(\tau, t) = 0$  if  $t \notin \mathcal{B}$ . Peaks (maximum values) are then searched in the histogram  $h$ . We typically select 100 peaks and then apply non maximum suppression, i.e., peaks that are closer than 1 minute to a stronger one are discarded. For each peak identified by  $\delta$ , the boundaries of the matching segments are then identified by collecting all the matches  $(\tau, t)$  associated with a hypothesis  $\delta$  such that  $|\tau - t - \delta| < 10$ . The final score is the sum of scores of these matches.

## 4.2 Handling the Temporal Burstiness

As mentioned in Section 3, consecutive video frames are often very similar, and so are their descriptors. This temporal dependency biases the scores returned by the Hough histogram, as bursts of votes occur for some frames, both on the query and database. This emphasizes them, i.e., they gather an abnormally large amount of scores.

We address this problem by modifying the scoring strategy in a way that mitigates this effect, in the spirit of the re-weighting scheme proposed in [6]. This is done by updating the score, prior to the Hough histogram computation in two steps:

$$s_1(\tau, t) = s(\tau, t) / \sqrt{\sum_{\tau \in \mathcal{Y}} s(\tau, t)} \quad \text{and} \quad s_2(\tau, t) = s_1(\tau, t) / \sqrt{\sum_{t \in \mathcal{B}} s_1(\tau, t)}, \quad (13)$$

where the computation is done efficiently by considering only the non-zero score values in the summations. The updated score  $s_2$  is used instead of the original scores in Equation 12. We will show in Section 5 that this procedure significantly improves the quality of the Hough estimation.

# 5 Experiments

## 5.1 Datasets and Evaluation Protocol

**Trecvid’08.** This dataset contains 200 hours of Dutch television broadcasts. It was used for the copy detection pilot task in the TRECVID’08 evaluation campaign. A set of 134 query clips was extracted from the dataset and 67 clips from other videos were added as negatives, i.e., with no corresponding videos in the database. Some clips were embedded in a distractor video and all were then transformed with 10 different transformations, see Table 2. As a result, 2010 queries with varying degrees of difficulty are used to evaluate a system.

The performance measure used to evaluate the search quality in the TRECVID competition is the *Normalized Detection Cost Ratio* (NDCR), which integrates

the cost of missing a true positive with the cost of retrieving an incorrect video. It is equal to 0 if all the true positives are returned before the false positives (no matter how many there are) and lower values of the NDCR correspond to better results. A result video segment is considered as a true positive if it overlaps with the ground-truth. We have used this measure<sup>3</sup> to compare our results with those obtained by the participants of the TRECVID'08 evaluation, see Subsection 5.3.

**STV** (Small TRECVID). In order to evaluate the parameters of our approach, we created a reduced version of the TRECVID dataset, referred to in the following as STV. It uses a subset of 21 h of the videos. From these videos we extracted a set of 100 clips not used by in the TRECVID'08 queries, embedded them in an independent distractor set and transformed them with some of the most challenging transformations.

This dataset is smaller than the TRECVID'08 dataset. However, the transformations are more challenging on average. We, therefore, obtain comparable conclusions with reduced runtime. Furthermore, using this dataset for parameter evaluation avoids optimizing parameters on the TRECVID query set, and provides a fair comparison with the state of the art on this dataset.

**YouTube.** In order to evaluate the scalability of our system, we have collected a dataset of YouTube videos. We downloaded a total of 38,000 videos from YouTube, corresponding to 189 million frames (or 2100 h). Most of the videos have a similar resolution as the TRECVID ones (about 352\*288) and the number of interest points extracted per frame is similar (about 300 per frame on average). These videos are used as distractors in our large scale experiment in Section 5.4.

**Evaluation measures.** In addition to the NDCR measure used for TRECVID, we have used two additional measures to evaluate performance, localization accuracy and average precision. The localization accuracy for a result segment is measured as the overlap between the ground-truth and the returned result:  $\Omega = |T_{\text{gt}} \cap T_{\text{found}}| / |T_{\text{gt}} \cup T_{\text{found}}|$ . If the match is incorrect,  $\Omega = 0$ , and if the localization of a match is perfect,  $\Omega = 1$ . Better matches have a higher overlap.

We have used the overall Average Precision (AP) as a quality measure. The results returned for all queries are evaluated together ranked by their scores. A match is considered a true positive if the overlap measure is above 0.5. A precision-recall curve evaluates the results as a function of the score. The area under this curve is the AP measure.

## 5.2 Impact of Parameters

Unless stated otherwise, for the parameters introduced in Sections 2 and 3, we have used the following values:

Descriptor	$D = 128 \times k = 8192$	$d = 2048$	$r = 1/10$
Coarse quantizer	$L = 2048$		$k_c = 64$
Fine quantizer	$L_f = 256$	$m_f = 64$	$k_f = 128$
Refinement	$L_r = 256$	$m_r = 16$	$k_r = 32$

<sup>3</sup> NIST (the institute organizing TRECVID) provided the software to compute this measure as well as the results of the other participants.



**Table 1.** Evaluation of memory usage, search accuracy and timings on the STV dataset. The method “Levels 1+2,  $r = 1$ ” indicates that frames are directly considered as segments in our method without any further refinement. The timings are given as a slowdown factor w.r.t. the “real” video time for one single 2.4GHz processor core (not including the frame description time).

structure/algorithm	total mem	$C_{\text{mem}}$	$C_{\text{dist}}$	AP	time
none/brute-force search	62 GB	62 GB	19 M	96.7	$89 \times$
Levels 1+2, $r = 1$	122 MB	25 MB	18432	90.1	$2.52 \times$
Levels 1+2, no refinement	12 MB	2.71 MB	18432	74.1	$1.10 \times$
Levels 1+2+3 ( $m_r = 16$ )	43 MB	2.73 MB	19710	91.2	$1.33 \times$
Levels 1+2+3 ( $m_r = 32$ )	73 MB	2.75 MB	19710	90.5	$1.42 \times$
Levels 1+2+3 ( $m_r = 64$ )	134 MB	2.79 MB	19710	91.7	$1.43 \times$

In the following, we measure the impact of these parameters. The performance is reported for the STV dataset.

**The dimensionality reduction** is evaluated on the two first levels of the method, i.e., without frame grouping ( $r = 1$ ) nor refinement (Levels 1+2 only). Our aggregation method is compared with PCA to reduce the  $D = 8192$  dimensions of the frame descriptor to  $d = 2048$ . For this operating point, dimensionality reduction with an aggregator (see Section 2.3) gives AP=87.7, and the PCA-based dimensionality reduction achieves AP=90.1. In the following, we use the PCA-based dimensionality reduction.

### Indexing: impact of the quantization and of the refinement step.

Table 1 shows the influence of the descriptor quantization and frame grouping on the search accuracy, memory usage and search time. Brute-force search gives an upper bound on the performance that can be achieved by using our frame descriptor, but is unreasonably expensive. We denote by *Levels 1+2* the methods that do not refine the segment level representation on the frame level. If  $r = 1$ , then frames are directly considered as segments, while *Level 1+2, no refinement* has the same effect as a subsampling, except that the average frame descriptor over a segment is used instead of a particular frame of this segment. This variant provides lower search quality, but is interesting to index very large datasets. One can observe that subsampling the video and indexing subsampled frames strongly degrades the performance, by 16 points of AP. The refinement improves the results. Short codes ( $m_r = 16$  bytes) are sufficient to capture most of the possible improvement. Note that, for large databases, this last refinement stage (Level 3) is the limiting factor in terms of memory usage, even with the setting  $m_r = 16$ .

**Burstiness handling.** We have evaluated the impact of our vote regularization procedure which addresses the problem of burstiness (cf. section 4.2). This method significantly improves the results, as shown below:

bursts regularization	AP
none	83.8
database-side (using $s_1$ in Equation 1.3)	84.3
full regularization ( $s_2$ )	91.2

**Table 2.** Evaluation of our method relative to other TRECVID’08 participants. The score is NDCR (the lower the better). The rank is obtained by taking the best run from each of the 22 participants.

no	transformation	best	second	ours	rank (/23)
1	camcording	0.079	0.363	0.224	2
2	picture in picture	0.015	0.148	0.321	4
3	insertion of patterns	0.015	0.076	0.079	3
4	strong re-encoding	0.023	0.095	0.064	2
5	change of gamma	0.000	0.000	0.023	3
6	photometric attacks	0.038	0.192	0.064	2
7	geometric attacks	0.065	0.436	0.140	2
8	3 random transformations from 6/7	0.045	0.076	0.437	5
9	5 random transformations from 6/7	0.038	0.173	0.693	5
10	5 random transformations	0.201	0.558	0.537	2

### 5.3 Comparison with State of the Art

Table 2 compares the NDCR scores of our system with the best and second best run (from different participants) of the TRECVID’08 competition<sup>4</sup>. We also provide the rank associated with our score. One can see that our system is very competitive, in particular for the most interesting transformations encountered on a peer-to-peer network: camcording and re-encoding. Note that the best score for these transformations is obtained with the approach of [6]. This approach requires 10 GB of RAM against 300 MB used here, i.e., it is difficult to scale to very large video sets. Furthermore, it is more than 5 times slower than the approach presented in this paper ( $13\times$  “real time” against  $2.47\times$  here on a single computing core).

We also compare our localization accuracy with the four best runs (from different participants) of the TRECVID’08 competition. We measure the accuracy on the 195 videos that were correctly retrieved by all 4 runs as well as by our method (i.e. the resulting segment has an overlap with the the ground-truth above 0.5). The measure used is average overlap. The results are:

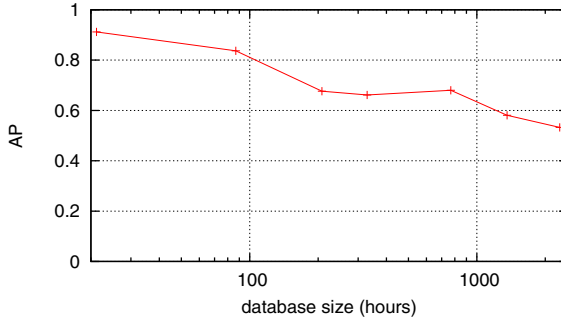
rank of the participant	1	2	3	4	ours
mean overlap	0.952	0.858	0.846	0.884	0.973

We can observe that our approach localizes the segments very precisely, i.e., with a better precision than the competing approaches.

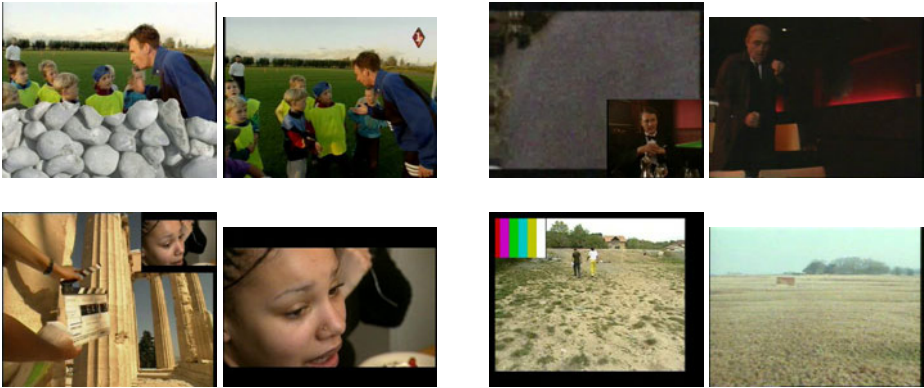
### 5.4 Large Scale Experiment

The large scale experiments are performed on the YouTube dataset merged with our STV dataset. Figure 3 shows the AP obtained as the function of the growing database size (up to 2316 hours). When performing the experiments on the whole

<sup>4</sup> The competitors can not be identified by name due to the non disclosure agreement of TRECVID.



**Fig. 3.** Retrieval performance on the STV dataset combined with a varying number of videos from the remaining TRECVID videos and YouTube



**Fig. 4.** Example results for video retrieval in our large scale dataset, (left) query and (right) best retrieved video. Left pairs: correct retrieval results. Right pairs: incorrect retrieval. Note the visual similarity between the queries and the retrieved videos.

set, the index requires 4.6 GB of RAM to index 208 million frames. The search is slower on that scale: 23.5 real time for a single processor core. One can observe that the AP measure decreases as to be expected, but that results are still good, i.e., we obtain an  $AP=0.53$  on the entire set. Typical retrieval results of our system are shown in Figure 4.

## References

1. Over, P., Awad, G., Rose, T., Fiscus, J., Kraaij, W., Smeaton, A.: Trecvid 2008-goals, tasks, data, evaluation mechanisms and metrics. In: Trecvid (2008)
2. Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., Gouet-Brunet, V., Bouje-maa, N., Stentiford, F.: Video copy detection: a comparative study. In: CIVR, pp. 371–378. ACM, New York (2007)

3. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
4. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1615–1630 (2005)
5. Joly, A.: New local descriptors based on dissociated dipoles. In: *CIVR* (2007)
6. Douze, M., Jégou, H., Schmid, C.: An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Transactions on Multimedia* 12, 257–266 (2010)
7. Perronnin, F., Dance, C.R.: Fisher kernels on visual vocabularies for image categorization. In: *CVPR* (2007)
8. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: *CVPR* (2010)
9. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: *ICCV*, 1470–1477 (2003)
10. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010)
11. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. *International Journal of Computer Vision* 60, 63–86 (2004)
12. Heikkilä, M., Pietikainen, M., Schmid, C.: Description of interest regions with local binary patterns. *Pattern Recognition* 42, 425–436 (2009)
13. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. *Computer Vision and Image Understanding* 110, 346–359 (2008)
14. Winder, S., Hua, G., Brown, M.: Picking the best Daisy. In: *CVPR* (2009)
15. Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., Girod, B.: Chog: Compressed histogram of gradients: A low bit-rate feature descriptor. In: *CVPR* (2009)
16. Calonder, M., Lepetit, V., Fua, P., Konolige, K., Bowman, J., Mihelich, P.: Compact signatures for high-speed interest point description and matching. In: *ICCV* (2009)
17. Perronnin, F., Liu, Y., Sanchez, J., Poirier, H.: Large-scale image retrieval with compressed Fisher vectors. In: *CVPR* (2010)
18. Nistér, D., Stewénus, H.: Scalable recognition with a vocabulary tree. In: *CVPR*, pp. 2161–2168 (2006)
19. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: *CVPR* (2007)
20. Yeh, M.C., Cheng, K.T.: Video copy detection by fast sequence matching. In: *CIVR* (2009)
21. Jégou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: *CVPR* (2009)

# Modeling the Temporal Extent of Actions

Scott Satkin and Martial Hebert

Carnegie Mellon University, The Robotics Institute  
{ssatkin,hebert}@ri.cmu.edu

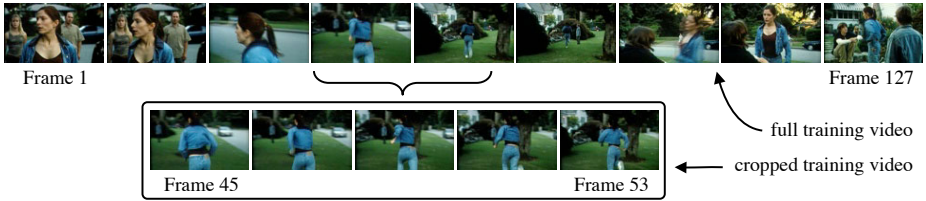
**Abstract.** In this paper, we present a framework for estimating what portions of videos are most discriminative for the task of action recognition. We explore the impact of the temporal cropping of training videos on the overall accuracy of an action recognition system, and we formalize what makes a set of croppings optimal. In addition, we present an algorithm to determine the best set of croppings for a dataset, and experimentally show that our approach increases the accuracy of various state-of-the-art action recognition techniques.

## 1 Introduction

There exists an inherent ambiguity for actions – When does an action begin and end? Unlike object boundaries in static images, where one can often delineate the boundary between an object and its background, determining the temporal extent of an action is often subjective. Consider the action “eating.” What is the precise moment that someone begins eating? When food is placed on a table? When a person picks up a fork? Moreover, when does the action end?

The problem is that the performance of an action recognition system may vary tremendously depending on the temporal boundaries chosen for the training samples. Researchers commonly crop training videos qualitatively based on the semantic definition of an action (which Cour *et al.* [2], Laptev *et al.* [3] and others point out can be a very difficult task). On the contrary, we set out to automatically determine temporal croppings for videos which optimize the performance of an action recognition system. Our objective is to identify the portions of each training video in a dataset, such that if an individual video is made any shorter, it would not fully capture the true essence of the action being performed. Conversely, if a cropped video is lengthened, it would add noise to the data, making the video less discriminative.

In this paper, we formalize what makes a cropping optimal with respect to the accuracy of a trained classifier, and we present an algorithm which identifies these discriminative portions of videos. Our strategy of temporally cropping training videos is applicable no matter what representation of an action is used. Therefore, we study the effect of our method on a diverse set of action representations, and show that on a wide variety of datasets we can consistently improve the performance of a classifier by temporally cropping training videos to their most discriminative portions.



**Fig. 1.** The most discriminative portion of a training video is automatically extracted. The cropped training video unambiguously belongs to the action category “running” from [1].

Figure 1 illustrates the concept of cropping a training video to its most discriminative portion. By running our algorithm on a video from the Hollywood-2 Human Actions and Scenes Dataset [1], we can automatically determine which portion of the video is best for detecting the “running” action. Note that unlike the original video from the dataset which contains ambiguous frames, our cropped video clearly depicts the action and disregards the frames which are not discriminative.

Collecting these types of videos, annotating their actions and delineating their boundaries is a labor-intensive task. For sufficiently large datasets, it is often impractical to do this manually. This process has been a focus of many research groups in recent years. In [1], [2], [3] and [4] the authors leverage the availability of movie scripts and closed captioning to get a rough idea of when actions occur in movies or television shows. The authors then employ various structured learning approaches to delineate these actions from their videos. Other work such as [5] and [6], focus on assisting users in the painstaking task of delineating the exact time and location of actions in videos.

Thus, it is impractical to label examples for supervised training by enforcing strict definitions of the temporal cropping of actions. Instead, our model for training involves taking video samples with approximate boundaries, and refining the samples during training. Moreover, since the temporal extent of an action is not a well-defined concept, we show that existing datasets can be further cropped during training to create a more discriminative set of training samples which improve the accuracy of a classifier, irrespective of what representation of human actions is used.

To show the broad applicability of our algorithm, we use four unique action representations: volumetric features, histograms of oriented gradients (HOG), histograms of optic flow (HOF) and point-trajectory features (Trajectons), which are a representative sampling of all major approaches. For each of these representations, we empirically show that identifying the most discriminative portions of each training video, and training a classifier on only those portions, improves overall performance.

## 2 Related Work

Our specific problem of temporally localizing the most discriminative portions of an action can be modeled with multiple instance learning, first explored by Dietterich *et al.* [7]. Recent work has demonstrated the importance of localizing or segmenting objects from static images for the task of recognition (*e.g.*, [8], [9] and [10]). Similar methods do apply, with the key distinction that we are dealing with a single interval on the temporal axis rather than a region in the image. Buehler *et al.* [11] applied multiple instance learning in the temporal domain with the unique goal of isolating individual exemplars for actions (sign language gestures). Our effort however is focused on improving classifier performance, not finding exemplars.

Recently, there have been a few attempts to mine action recognition datasets to solve this problem. In [12], Nowozin *et al.* present an algorithm which searches for discriminative subsequence patterns in videos. However, since there is no constraint that the subsequences be continuous, this solution is equivalent to finding individual space-time features in the video which are discriminative, as opposed to our algorithm which determines the most discriminative portion of each video. Yuan *et al.* [13] propose a branch-and-bound algorithm which searches for a 3-D bounding box, akin to our temporal cropping, by maximizing mutual information of features and actions under a naïve Bayes assumption. Their method is specific to an STIP action recognition model, and cannot be applied to other systems. However, our algorithm treats the underlying action recognition system as a black box and only requires the ability to train on a subset of the dataset and evaluate its precision.

Most related to our work is that of Duchenne *et al.* [4]. Their work aims to automatically find the location of actions in videos, in a semi-supervised manner. By leveraging the availability of movie scripts and subtitles, their system begins with a rough estimate of when an action occurs. The authors then refine the location of this action using structured learning. A key distinction is that their goal is to determine temporal boundaries that approximate the way a human would qualitatively crop the data. On the contrary, our algorithm directly optimizes the accuracy of a classifier trained using the cropped videos. Unlike the authors of [4], who strive to generate croppings which perform as well as human-labeled data, we consider the performance of a classifier trained on manually cropped actions to be a baseline. Thus, we can take training data such as [4]’s “ground-truth” croppings, and further refine the temporal boundaries to produce a classifier that outperforms human-labeled data on the task of action recognition.

## 3 Problem Formulation and Overall Approach

We define an “optimal set of croppings” as the set of start  $f^0$  and end  $f^1$  frames for each video  $\mathcal{F}_i$  of class  $\mathcal{C}_i$  in our training dataset which produces a classifier

with the highest leave-one-out training accuracy. This can be quantified with the following high-level equation:

$$\operatorname{argmax}_{\{v_i:(f_i^0, f_i^1)\}} \sum_{i=1}^n \operatorname{classify}(\operatorname{train}(\mathcal{F}_{(1\dots n)\neq i}, f_i^0, f_i^1), \mathcal{F}_i) = \mathcal{C}_i. \quad (1)$$

For  $n$  training videos, each with  $|f|$  frames, there are  $O(n^{|f|^2})$  possible sets of temporal croppings (in [4],  $n = 823$  and  $|f| \approx 280$ ). Due to this exponentially high-dimensional search space, it is intractable to test the accuracy of a classifier trained on all possible sets of croppings. Thus, a major question we address is: *How can we optimize over the massive set of potential croppings?*

In this paper we leverage the fact that portions of videos which are most confidently and correctly classified by a trained action recognition system are highly correlated with actions of the same class and differ from actions of other classes. Therefore, these portions of the videos are discriminative and are a good choice for training our classifier.

Our overall approach to determine a good cropping for an individual training video is as follows:

1. Split the video we aim to crop into its  $|f|^2/2$  possible temporal croppings.
2. Train a classifier on the remaining training videos, excluding the one from step 1.
3. Evaluate this classifier on each of the  $|f|^2/2$  croppings.
4. Select the individual cropping that was correctly classified with the highest level of confidence.

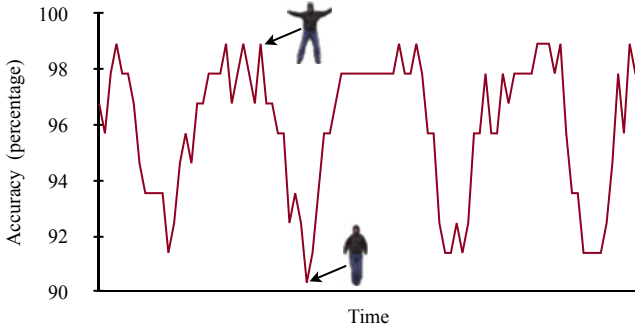
This approach treats the underlying action recognition system as a black box; thus, it can be applied to almost any classifier. It is a well-founded solution, which takes the form of stacked generalization [14]. Depending on the specific type of action representation being used, there are different considerations which must be taken involving tractability and the overall method of classification. Sections 4 and 5 explore two instances of this general approach: one based on space-time representations using volumetric features, the other using a more common bag-of-words representation.

## 4 Proof of Concept Experimentation

We begin by evaluating the effectiveness of our approach using Ke *et al.*'s volumetric features action recognition model [15]. This algorithm creates an action model from a single training video by segmenting a person from their background in each frame to create a 3-D silhouette. Detection is performed by comparing the boundary of this 3-D template to the edges of over-segmented frames from a testing video. We chose to experiment on Ke *et al.*'s volumetric features in this section, as a representative sample of space-time action models; although, our method can easily be applied to similar algorithms such as Rodriguez *et al.*'s "Action MACH" or Shechtman *et al.*'s "Space-Time Behavior Based Correlation" techniques [16] [17].



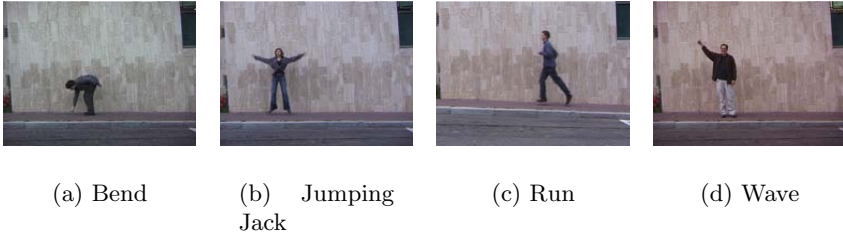
Since [15]’s approach builds an action recognition model from a single video, as opposed to many videos, there is only a quadratic number of croppings to consider, as opposed to the super-exponential number of possible croppings when training on multiple videos. Additionally, because the template comparison performed in [15] approximates a convolution operation, which is commutative (i.e., the template and training video can be swapped), our methodology of running a training video through a classifier as if it were a testing video to efficiently identify discriminative portions is a theoretically well-founded approximation to the high-dimensional optimization problem.



**Fig. 2.** Plot showing the accuracy of volumetric feature templates extracted from different times for an instance of the “jumping jack” action from the Weizmann Actions as Space-Time Shapes Dataset [18].

To quantify the benefits of temporally cropping a training video prior to creating an action recognition model, we begin by brute-force testing the accuracy of fixed length templates centered at every frame of a training video from the Weizmann Actions as Space-Time Shapes Dataset [18]. Figure 2 shows how the accuracy of a model varies based on what portion of a video it is extracted from. The periodic nature of the “jumping jack” action is quantifiable from the sinusoidal shape of the accuracy plot. A one-frame template is shown for the most discriminative and least discriminative portions of the video. It is intuitive that the most discriminative part of a jumping jack is when a person is in mid-air with all limbs extended outwards; on the contrary, when the person lands, they are momentarily indistinguishable from a person standing still. Note that the accuracies of the models vary between 90% and 99%, indicating that there is much to be gained by intelligently cropping training videos.

Table 1 reports the results of our approach on the entire Weizmann Actions as Space-Time Shapes Dataset [18] (shown in Figure 3) which contains 10 action classes. An interesting observation is that some classes such as “bend” and “jumping jack” have a large gap between the best and worst temporal croppings; however, actions like “run” and “walk” have less room for improvement. This is intuitive since an action like “bend” occurs at a specific instance, while “walk” has a relatively consistent appearance over time. The average accuracy of all 10 classes is also reported.



**Fig. 3.** Example categories from the Weizmann Actions as Space-Time Shapes Dataset [18].

**Table 1.** Effects of cropping the Weizmann Dataset using Ke *et al.*'s volumetric feature action recognition model.

Action	Worst Cropping	Best Cropping
	Accuracy	Accuracy
Bend	90.63	98.00
Jumping Jack	90.94	97.70
Run	93.39	96.47
Walk	93.55	95.70
10-class Average	91.98	95.76

## 5 Temporal Refinement of Videos Using a Bag-of-Words Approach

Datasets such as the KTH dataset [19], and the Weizmann dataset [18] used in Section 4 have been criticized in recent years for not being a realistic sampling of actions in the real world. To tackle more complex datasets, researchers have extended the bag-of-visual-words technique from object recognition in images into the temporal domain. For example, [1], [3] and [4] represent actions as histograms of space-time interest points (STIPs), which encode both static image gradients and optic flow. The authors of [20] and [21] propose a similar method to STIPs-based systems; however, their features are based on the trajectories of tracked interest points, which can better model motion than the optic flow vectors used in previous work.

Therefore, in this section we show how our method from Section 3 can be used to determine sets of training video croppings which improve the accuracy of these bag-of-words based classifiers, which are a representative sampling of state-of-the-art techniques. We evaluate the effectiveness of our approach on two challenging datasets: Messing *et al.*'s University of Rochester Activities of Daily Living [20] and Marszałek *et al.*'s Hollywood-2 Human Actions and Scenes Dataset [1].

## 5.1 Determining the Best Croppings

To search for the best set of croppings for each video in our training set, we augment the traditional SVM classification formulation as follows:

$$\underset{\{\forall i: (f_i^0, f_i^1)\}, \mathbf{w}, b, \xi}{\operatorname{argmin}} \left( \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right), \quad (2)$$

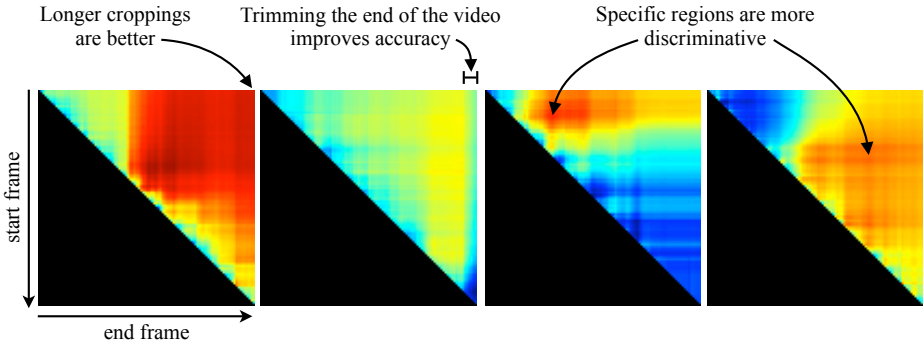
$$\text{subject to: } \forall i: y_i \left( \mathbf{w} \cdot \phi \left( \sum_{f=f_i^0}^{f_i^1} H_i(f) \right) + b \right) \geq 1 - \xi_i. \quad (3)$$

Our max-margin formulation minimizes over  $f^0$  and  $f^1$ , the starting and ending frames for each training video (defined in Section 3), in addition to the other standard SVM parameters. The constraints in Equation 3 include a histogram accumulation of features between start and end frames.  $H_i(f)$  denotes the histogram of the quantized features from frame  $f$  of video  $i$ . As per 3 and 21, we use 4000 histogram bins for HOF and HOG space-time interest points, and 512 bins for Trajection features. All feature vectors are  $\mathcal{L}^1$  normalized prior to SVM training or classification. For consistency and simplicity, we use the same  $C$  value and a linear kernel for all experiments in this section.

Since it is infeasible to solve this high-dimensional integer linear program, we will focus on detecting the most discriminative portion of each video individually, using the approach we introduced in Section 3. This is done by training a multiclass SVM on all uncropped training videos, excluding the one which we aim to crop. We use Wu *et al.* 22’s method of multiclass SVM classification which not only assigns category labels, but also estimates the probability that an instance belongs to each of the classes. We then evaluate the SVM on the  $|f|^2/2$  possible temporal croppings of the video excluded from training to determine how discriminative each segment of the video is.

Figure 4 includes visualizations which indicate what portions of individual videos from the Hollywood-2 Dataset 4 are most discriminative. Each pixel in these images represents a different cropping (blue pixels indicate the least discriminative croppings, and red signifies the most discriminative croppings). The vertical and horizontal axis specify the starting and stopping frames ( $f^0$  and  $f^1$ ), respectively. The lower left portion of each of these figures is blank, since these are invalid croppings (i.e.,  $f^1 < f^0$ ). The upper right corner represents the full uncropped video. These experiments were conducted using Laptev *et al.*’s spatio-temporal feature extraction tool 3, and LIBSVM 23.

The leftmost heatmap in Figure 4 was generated from a video in which the entire video contributes to its overall discriminative quality. The farther from the diagonal  $f^0 = f^1$ , the longer the cropping, and the more discriminative the video gets. For videos like this one, we cannot improve the classifier accuracy with temporal cropping. On the contrary, the next heatmap shows a fascinating property inherent to many videos. The rightmost portion of the heatmap has a distinct vertical cyan stripe. This indicates that there are unusual features



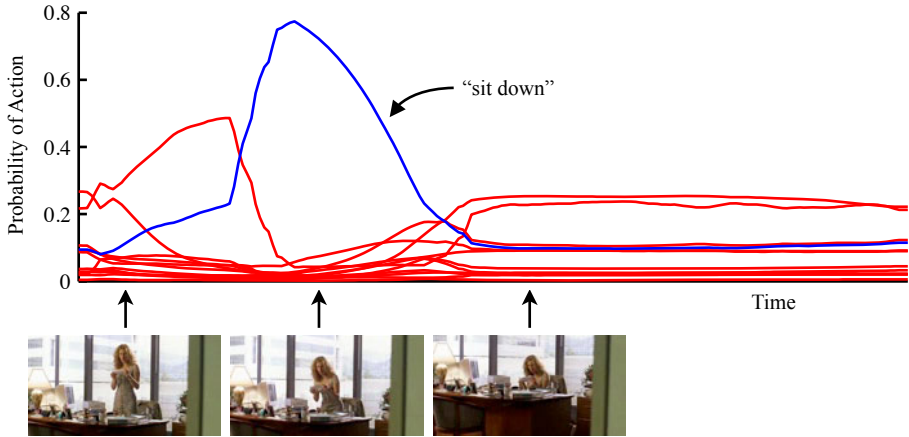
**Fig. 4.** Heatmaps showing which croppings are most discriminative for individual videos from the Hollywood-2 Dataset [4].

contained in the final frames of the video, which make the video far less discriminative. However, by trimming the last frames off of the video, independent of the starting frame, the classifier’s performance can increase. Lastly, the two rightmost heatmaps in Figure 4 depict videos where most croppings would make a bad model for action recognition. For videos of this nature, it is important to choose a cropping from within the specific region we identify to be discriminative.

It is combinatorially intractable to optimize over all  $O(n^{|f|^2})$  possible combinations of temporal croppings for a dataset with  $n$  training videos. Therefore, we impose the constraint:  $\forall i, (f_i^1 - f_i^0) / |f_i| = \alpha$ , which restricts our search space to the set of cropped video clips which are the same fixed percentage  $\alpha$  of their full version. It is intuitive that if  $\alpha$  is too low, we are throwing away too much of the training data; conversely, if  $\alpha$  is unnecessarily high, we are not sufficiently cropping the training data to achieve the best possible results. Therefore, we run cross-validation to identify the ideal value of  $\alpha$  for each dataset.

This process begins by randomly splitting the training data into two parts: a training set and a validation set. Heatmaps are generated for all videos in the training set using the leave-one-out method described above. Using these heatmaps, for a given value of  $\alpha$  (which corresponds to a diagonal line in each heatmap), we can pick the most discriminative cropping. We then iterate over all values of  $\alpha$  from 1% to 100%, and train a classifier on the set of croppings which corresponds to each particular value of  $\alpha$ . The validation set of data that was withheld can now be used to determine the best value of  $\alpha$ . By decoupling the location and length parameters of the video segments we aim to extract, we can efficiently identify discriminative combinations of cropped training videos. This approximation scheme (which requires no parameter tuning) yielded good results on all of the datasets with which we experimented.

Our algorithm scales linearly with the dataset size, and is parallelizable. Additionally, our algorithm acts as a pre-processing step which only needs to be run once prior to training. Therefore, computational expense is not a major factor.



**Fig. 5.** The predicted probabilities for each of the 12 action classes as a function of time for a “sit down” video from [1]. The correct label is indicated in blue. Frames from the video are shown on the x-axis at their timestamp.

## 5.2 Classification via Detection

The standard classification paradigm of training on one set of videos, and classifying another set of videos, is not a representative problem. It is unrealistic to expect that a real-world application which uses an action recognition system would have well-cropped test data, where each video is trimmed to the length of an action. This has a major impact on how performance is evaluated (which we discuss in Section 6). Therefore, rather than extracting features from each video in its entirety, and classifying whole test samples, we employ a methodology which essentially *detects* the occurrence of an action in each video. This paradigm does not require the test videos to be cropped to the temporal extent of an individual action and can easily be altered to run on a stream of data.

For each testing video, we evaluate the multiclass SVM on a sliding window of frames. The duration of the sliding window can be set to the median length of the cropped training videos to achieve good results. We further improve the accuracy (on the order of 1%) using cross-validation to tune this parameter. During classification, we extract features from each set of frames in the sliding window, and consider each of these segments of video independently of the test video as a whole. Using [22]’s method, the probability that each of these segments belongs to individual action classes is determined.

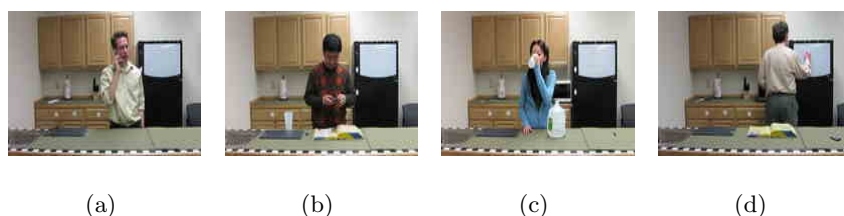
Figure 5 shows how predicted category labels can vary drastically depending on the portion of the test video used for classification. In this example, the action “sit down” occurs almost instantaneously at approximately one-third of the way through the video (as shown in the video frames below the x-axis). For this brief portion of video, the event “sit down” (indicated in blue) is predicted



**Fig. 6.** Example categories from the Hollywood-2 Human Actions and Scenes Dataset [1]. Pictured left to right: Drive, Kiss, Hug, Answer Phone.

**Table 2.** Effects of cropping the Hollywood-2 Dataset using three different action representations.

	Baseline Accuracy (using full videos)	Our Accuracy (cropped videos)	Absolute Change cropped - full	% Improvement (cropped - full)/full
Trajectons	37.84	41.85	4.01	10.60
HOG	33.08	33.71	0.63	1.90
HOF	38.47	43.48	5.01	13.02



**Fig. 7.** Example categories from the University of Rochester Activities of Daily Living [20]. Pictured left to right: Answer Phone, Dial Phone, Drink Water, Write on Board.

**Table 3.** Effects of cropping the University of Rochester Dataset using three different action representations.

	Baseline Accuracy (using full videos)	Our Accuracy (cropped videos)	Absolute Change cropped - full	% Improvement (cropped - full)/full
Trajectons	46.00	54.00	8.00	17.39
HOG	54.67	60.00	5.33	9.75
HOF	79.33	80.00	0.67	0.84

by the SVM with high-confidence. At the beginning and end of the video, when the actress is either standing or has already finished sitting down, the classifier picks one of the other 12 classes (indicated in red), with a significantly lower confidence. Because we want to evaluate the performance of our algorithm

in the context of a classification task, we assign a single label to an entire testing video by simply taking the peak response of the SVM classification from all timestamps.

### 5.3 Experimental Analysis

To demonstrate the broad applicability of our approach, we evaluate the benefits of temporally cropping videos using the method described in Section 5.1 on three unique action representations: Histograms of Optic Flow (HOF) [3], Histograms of Oriented Gradients (HOG) [3] and Trajectons [21].

Our goal is to empirically show that: *By adjusting the temporal boundaries of training videos as a pre-processing step, we can improve the accuracy of a classifier, regardless of the action representation being used.* The sole purpose of the experimental analysis is to evaluate the *added benefit* of temporally cropping training videos to their most discriminative portions.

To compare our work with other action classification papers, we can only train a single multi-class SVM (and therefore can only use one set of croppings). However, our solution is general and without modification to the algorithm, we could determine a separate set of croppings for each action to train individual SVMs.

Table 2 reports the improvements from cropping the Hollywood-2 dataset prior to training. For consistency with other experiments in this paper, we use overall percentage accuracy as our performance metric. The baseline accuracy is the performance of a classifier which is trained and tested using full videos from the dataset. We compare that to the accuracy of a classifier which is trained only on the most discriminative portions of a video, using our cropping algorithm. The final columns quantify the absolute and percentage improvements due to cropping. Similarly, Table 3 reports the improvements from cropping the University of Rochester dataset prior to training. The key observation is that our strategy consistently improves the performance of an action recognition system independent of what types of features are used.

## 6 Discussion and Future Work

This research has begun to explore the benefits of identifying the most discriminative portion of training videos. We presented a framework which uses a trained classifier to predict the most discriminative part of each video, irrespective of the action representation being used. Our methodology has proven to be broadly applicable, and shows the tremendous impact that the temporal cropping of videos has on the accuracy of an action recognition system. Future work will continue researching the effects of the identifying the most discriminative portions of training videos. We hope that combining multiple croppings and perhaps extending our approach to search for regions both spatially and temporally will yield even further improvement.

As a final note, while studying this problem, we noticed an important property inherent to many datasets. Because videos in most action recognition datasets

are cropped to the approximate temporal extent of each action, the length of each test sample tends to be highly correlated with its action label. For example, 38% accuracy on the University of Rochester Dataset and 27% accuracy on the Hollywood-2 Dataset can be achieved by classifying solely on the number of frames in each video. This bias can easily be exploited if care is not taken to explicitly normalize for this issue. For example, it is necessary to  $\mathcal{L}^1$  normalize feature histograms prior to training or classification. Not normalizing these feature vectors can lead to a substantial boost in classifier accuracy (*e.g.*, 15% increase in accuracy using Trajectons on the University of Rochester Dataset). Features themselves, such as those in [20], can also implicitly encode for the length of videos, by not limiting the number of frames which they describe.

Although using these types of features or not explicitly normalizing to ignore the number of frames in each video can yield better classification results, this is an artifact of the dataset biases and cannot be generalized to other action recognition tasks. As discussed in Section 5.2, it is not reasonable to assume that videos will be cropped tightly to the temporal extent of each action. For example, in the real-world problem of detecting the occurrence of actions in video streams, models which implicitly encode for the length of actions are no longer applicable. Moreover, if we knew how to crop these videos, this would be a solved problem. That is why we chose to implement classification as a detection problem.

This suggests ways to revisit the generation of datasets for action recognition to avoid these biases. By providing test data that is not cropped to the temporal boundaries of each action, it ensures that good action recognition systems are a result of understanding and modeling actions, not exploiting properties inherent to individual datasets.

## Acknowledgements

This work was supported in part by the Traffic21 initiative of Carnegie Mellon University and The Technology Collaborative. We would like to thank Ivan Laptev, Pyry Matikainen and Ross Messing for sharing their datasets, source code and feature extraction utilities, and Intel Research Pittsburgh for providing computing resources.

## References

1. Marszałek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
2. Cour, T., Jordan, C., Miltsakaki, E., Taskar, B.: Movie/script: Alignment and parsing of video and text transcription. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 158–171. Springer, Heidelberg (2008)
3. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
4. Duchenne, O., Laptev, I., Sivic, J., Bach, F., Ponce, J.: Automatic annotation of human actions in video. In: ICCV (2009)



5. Hall, J., Greenhill, D., Jones, G.A.: Segmenting film sequences using active surfaces. In: *ICIP* (1997)
6. Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., Cohen, M.F.: Interactive video cutout. *ACM Transactions on Graphics* 24, 585–594 (2005)
7. Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 31–71 (1997)
8. Malisiewicz, T., Efros, A.: Improving spatial support for objects via multiple segmentations. In: *BMVC* (2007)
9. Uijlings, J.R.R., Smeulders, A.W.M., Scha, R.J.H.: What is the spatial extent of an object. In: *CVPR* (2009)
10. Pantofaru, C., Schmid, C., Hebert, M.: Object recognition by integrating multiple image segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008*, Part III. LNCS, vol. 5304, pp. 481–494. Springer, Heidelberg (2008)
11. Buehler, P., Zisserman, A., Everingham, M.: Learning sign language by watching TV (using weakly aligned subtitles). In: *CVPR* (2009)
12. Nowozin, S., Bakir, G., Tsuda, K.: Discriminative subsequence mining for action classification. In: *CVPR* (2007)
13. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: *CVPR* (2009)
14. Wolpert, D.H.: Stacked generalization. *Neural Networks* 5, 241–259 (1992)
15. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: *ICCV* (2007)
16. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In: *ICCV* (2008)
17. Shechtman, E., Irani, M.: Space-time behavior based correlation. In: *CVPR* (2005)
18. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: *ICCV* (2005)
19. Schuld, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: *ICPR* (2004)
20. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: *ICCV* (2009)
21. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: *VOEC Workshop* (2009)
22. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research* 5 (2004)
23. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software, Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

# Content-Based Retrieval of Functional Objects in Video Using Scene Context\*

Sangmin Oh, Anthony Hoogs, Matthew Turek, and Roderic Collins

Kitware Inc.

**Abstract.** Functional object recognition in video is an emerging problem for visual surveillance and video understanding problem. By functional objects, we mean objects with specific purpose such as postman and delivery truck, which are defined more by their actions and behaviors than by appearance. In this work, we present an approach for content-based learning and recognition of the function of moving objects given video-derived tracks. In particular, we show that semantic behaviors of movers can be captured in location-independent manner by attributing them with features which encode their relations and actions w.r.t. scene contexts. By scene context, we mean local scene regions with different functionalities such as doorways and parking spots which moving objects often interact with. Based on these representations, functional models are learned from examples and novel instances are identified from unseen data afterwards. Furthermore, recognition in the presence of track fragmentation, due to imperfect tracking, is addressed by a boosting-based track linking classifier. Our experimental results highlight both promising and practical aspects of our approach.

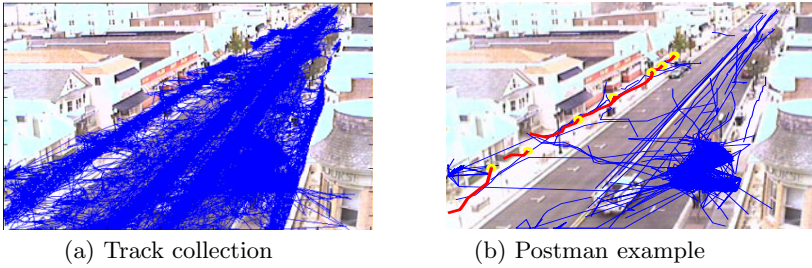
## 1 Introduction

Functional object recognition in video is an emerging problem for visual surveillance and video understanding problem. By functional objects, we mean objects with specific purpose such as postman and delivery truck, which are defined more by their actions and behaviors than by appearance. Yet, most object recognition algorithms attempt to classify objects based solely on appearance, largely because static imagery was the only data available. With the recent, explosive increase in video sensors, it is now possible to classify objects based on their movements and activities in applications such as surveillance and aerial videos.

In this work, we present an approach for content-based learning and recognition of the function of moving objects given video-derived tracks. As an example, Fig. II(a) shows subset of our dataset where 1,442 tracks are computed from an uncontrolled webcam video spanning 86 minutes (among total 7 days). Fig. II(b)

---

\* This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W31P4Q-09-C-0256. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.



**Fig. 1.** (a) Collection of 1442 tracks spanning 86 minutes of webcam video. (b) Successfully identified sequence of a postman in a scene. His trajectory is fragmented into seven red tracks where yellow circles mark the beginnings of tracks, and many other simultaneous tracks are nearby (blue).

shows an example of a postman sequence (in red) which was successfully recognized among concurrent tracks (in blue). The video was recorded from a webcam located in Ocean City, NJ, USA. The frame rate is mostly 1Hz or worse, and the pixel-level noise is very high, contributing to degraded tracking performance typical in many surveillance video data.

The postman example illustrates the major challenges for functional object recognition. First, functional models should capture location-independent behavioral semantics, because location variability can be substantial across examples and only limited number of training examples are often available for interesting objects. Imagine the case of a delivery truck. It can literally stop at any parking spots, and person from the truck may visit any store in a scene. In our knowledge, learning of location-independent functional object models is novel, and mostly unexplored territory. Another core challenge is that the trajectories of target objects, marked in red in Fig. 1(b), are often fragmented into multiple tracks, and many other tracks (blue) are nearby in the same time interval – each track is too short to characterize the function, so that we must link tracks to identify functions. Tracking errors are often innate. Tracking algorithms can miss objects entirely, lose an object after tracking it for a while, or virtually any combination of these. Trackers are often optimized to avoid identity switching errors, which will result in greater track fragmentation.

Our solution for location-independent semantic behavior learning is to incorporate *scene context*. By scene context, we mean local scene regions with different functionalities such as doorways and parking spots which moving objects often interact with. They are ‘contexts’ because they are surrounding information, providing additional cues about mover’s functions. Every track is encoded with Boolean features which capture its interactions w.r.t. scene contexts, e.g., the activity of people walking into roads can be characterized by attributes such as `move_on_sidewalk`, `move_towards_road`, and `move_on_road`. In particular, we explored the use of both manually and automatically obtained scene contexts. Based on these representations, functional models are learned from examples, and novel instances are recognized from unseen data. To model behavior

over time in the presence of track fragmentations, we have formulated a two-level modeling scheme. At the lower level, collected tracks are clustered based on features relating them to scene elements, resulting in elementary models corresponding to different categories of low-level behaviors such as “walking on sidewalk” and “crossing road”. This scheme is motivated by the observation that tracks tend to be fragmented when there are substantial changes in low-level activities. At the higher level, composite full functional object models are learned in a supervised fashion, using the elementary models as building blocks, abstracting away low-level information. In terms of modeling regimes, we have investigated three approaches: (1) unigrams, (2) bigrams, and (3) HMMs.

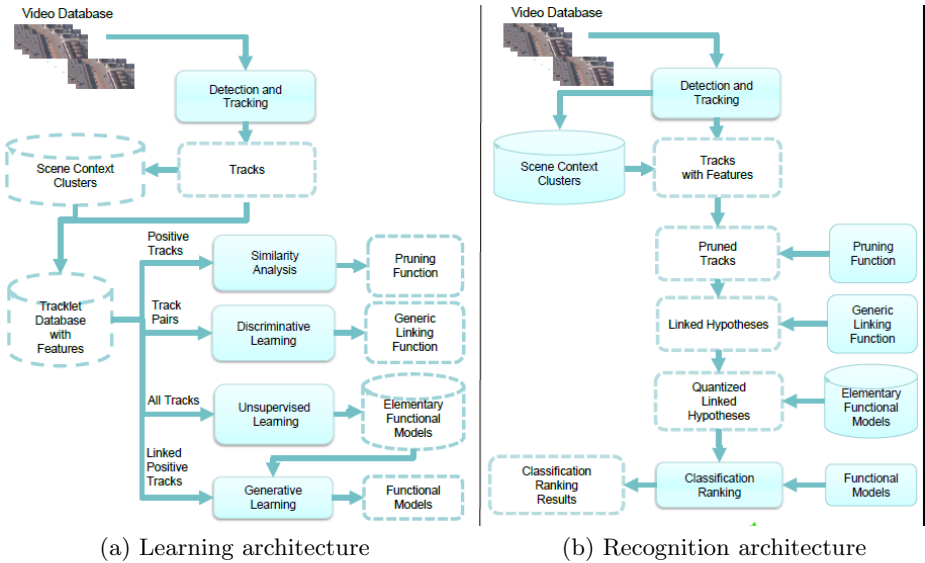
To address the fragmentation issues during recognition, we have developed a track linking classifier based on Adaboost.M1 [1]. The classifier computes link probabilities for every pair of tracks based on agreement between their features. Then, sequences of tracks with higher link probabilities are formed into functional behavior hypotheses to be evaluated against full functional object models. Additionally, a pruning scheme has been developed to filter out large portion of unrelated concurrent tracks prior to recognition, which leads to reduced number of hypotheses and alleviates computational demand substantially.

## 2 Related Work

The concept of functional object recognition for static objects was pioneered with work on recognizing chairs in static images [2]. Later, work has appeared on integrating observed human limb activities in video for static object recognition [3,4]. The notion of functional objects in this work goes beyond previous work in that they are actively moving entities which interact with environments.

Most work on trajectory analysis [5,6,7] focuses on grouping trajectories based on their locations and learning normalcy models. The resulting models are mostly location-dependent, primarily due to the adopted track features which heavily rely on either image or world coordinate information. In this work, we incorporate scene context features which largely abstract away location information. Accordingly, resulting models in our work depends less on location information and deliver interpretable semantics. A related work is [8] where the states of HMMs are semantic primitives such as **C**lose**T**o and **M**oving. In our work, however, semantic primitives are learned in an unsupervised manner.

The tracklet link classifier in this work differs from previous work [9,10] in that links are formed non-exclusively. The motivation is that we focus more on improving true positive retrieval ratios of links that belong to occurrences of functional objects, allowing multiple links to be formed from a track to other tracks, aiming to examine as many linked hypotheses as possible. The potential risk of exponential growth in number of formed hypotheses is compensated by pruning seemingly unrelated tracks prior to linking (see Sec. 6). Previous work addresses this issue by imposing exclusive linking constraints which is solved by global solutions such as Hungarian algorithm [9] or cost-flow network [10]. Our link classifier is learned within boosting framework with two noteworthy differences from hybrid boosting approach [10]. First, we directly learn binary link

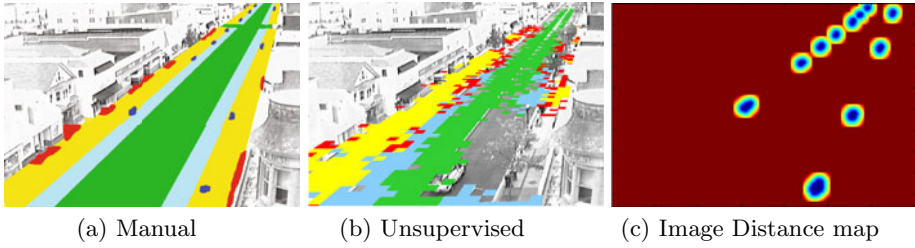


**Fig. 2.** (a) Architecture for learning functional object models. (b) Architecture for recognition. Solid boxes denote algorithmic processing units or existing information. Dashed boxes represent newly computed outcomes after every processing module.

classifier and do not need to adopt ranking-based methodology in [10], because we do not impose exclusive linking constraint through global solution in latter stages. Second, agreements between semantic Boolean features are used as crucial information for linking in our work, different from detailed kinematics and appearance features in [10]. In far-field videos, such features are less reliable due to the challenges of mover detection with accurate tight bounding boxes.

### 3 Overview of System Architecture

The overall architecture of functional model *learning* is illustrated in Fig. 2(a). Solid boxes denote algorithmic processing units or existing information. Dashed boxes represent newly computed outcomes, e.g., learned models, after every processing module. First, input video is stabilized and geo-registered (not shown), then object tracks are extracted using our tracking method which uses background subtraction and performs global multi-object tracking, similar to [9]. Note that our overall approach does not depend on the tracker, and in principle any video detection and tracking system could be used. For scene contexts (Sec. 4), manually labeled information can be used. Otherwise, scene contexts can be automatically identified. Then, tracklet database is formed where each track is attributed with comprehensive set of features, including its interaction with scene contexts (Sec. 4). Then, four major modules are learned from the feature-indexed database: (a) pruning function (Sec. 6), (b) a track linking function



**Fig. 3.** (a) Manual scene context labels: road (green), parking spots (light blue), sidewalks (yellow), building entrances (red), and trash cans (dark blue). (b) Unsupervised scene context learning results with four clusters. (c) An example distance map on image to trashcan context (in blue on the leftmost figure).

(Sec. 6), (c) shared elementary functional models (Sec. 5), and (d) full composite functional models of interest (Sec. 5). In particular, a set of elementary functional models are learned from a large pool of tracks, relations and actions using unsupervised clustering. Then, we learn full composite functional models of interest from a selected set of linked positive examples where the pre-computed elementary functional models are used as building blocks. A noteworthy advantage of the current architecture is that most of the intermediate computational results such as scene context clusters, generic linking function, and elementary functional models can be reused to learn new functional models.

In the *recognition* phase shown in Fig. 2(b), novel data are fed into the system either as video streams or a set of video clips from which object tracks are extracted. The same features and relations used in learning phase are computed for each track. Then, potentially irrelevant tracks w.r.t. the functional object under search are pruned using learned pruning function, leaving only promising ones for further processing. The survived set of tracks are linked via the learned generic linking function to yield linked hypotheses. Then, every linked hypothesis is quantized into a sequence of elementary functional models which will be scored w.r.t. a full functional model. Finally, either ranking methodology or detection thresholds are applied to suggest promising instances to operators.

## 4 Scene Context and Track Features

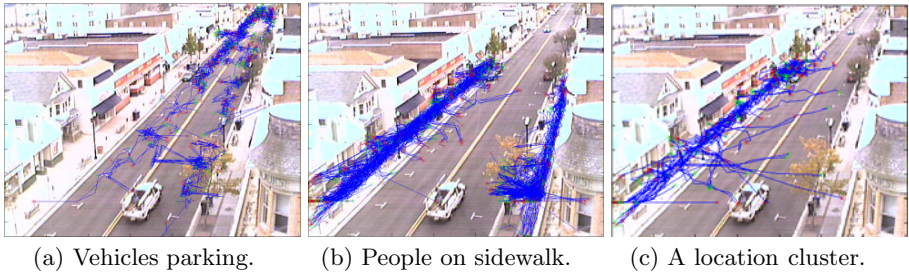
**Scene Contexts.** Once track(let)s are computed, the next level of representation is the characterization of tracks by relations between tracks and scene. In particular, the manner and timing of semantic interactions between a moving object and nearby static scene contexts can indicate significantly different types or behaviors. Fig. 3(a) illustrates five different types of manually identified and labeled scene contexts: road, parking spots, sidewalks, building entrances, and trash cans. Such manual information is becoming increasingly available through various geo-spatial databases, e.g., Google Maps. On the other hand, scene regions with different functionalities can be automatically grouped in an unsupervised manner. In this work, unsupervised scene contexts are obtained by

**Table 1.** List of track features in three categories: (1-7) track-level, (8-46) contextual, and (47-48) composite. The rightmost three columns indicate which features are used for latter stages of linking (L), track clustering (C), and pruning (P).

ID	Category	Type	Feature Description	L	C	P
1	Track	Continuous	2D initiating locations in world	o		
2	Track	Continuous	2D terminating locations in world	o		
3	Track	Continuous	2D initiating locations in image	o		
4	Track	Continuous	2D terminating locations in image	o		
5	Track	Continuous	2D average speed in world (m/s)	o		
6	Track	Continuous	Initiating time (in seconds)	o		
7	Track	Continuous	Terminating time (in seconds)	o		
8	Track	Boolean	Tracking bounding box size indicates person?	o	o	o
9	Track	Boolean	Tracking bounding box size indicates vehicle?	o	o	o
10	Track	Boolean	Fast moving (within normal vehicle speed)?	o	o	o
11	Track	Boolean	Slow moving (within normal human speed)?	o	o	o
12-26	Context	Boolean	Move on scene contexts within world?	o	o	o
27-31	Context	Boolean	Move nearby scene contexts within world?	o	o	o
32-36	Context	Boolean	Move nearby scene contexts within image?	o	o	o
37-41	Context	Boolean	Move away from scene contexts within world?	o	o	o
42-46	Context	Boolean	Move toward scene contexts within world?	o	o	o
47	Composite	Boolean	Possibly a human?	o	o	o
48	Composite	Boolean	Possibly a vehicle?	o	o	o

accumulating the intersecting track behaviors per region, and clustering them, similar to [11]. Additionally, it is worthwhile to note that functional scene region detectors can be trained in a supervised manner, and can be used to detect semantic concepts such as parking spots or doorways [12]. Fig. 3(b) shows the scene context clusters obtained through this approach where parameters were tuned to produce similar number of clusters. It can be seen that the results are fairly interpretable and similar to manual labels. Each unsupervised cluster delivers interpretations of road, sidewalk, parking spots, and short activity areas such as garbage cans and partial doorways. The experimental results in Sec. 7 report recognition results based on both manual and unsupervised scene contexts.

**Track Features.** Every track is attributed with three categories of features: (1) track-level, (2) contextual, and (3) composite. Track-level features consists of both continuous and Boolean features while the other two categories include only Boolean features. Manual camera calibration provided a homography mapping image locations to the ground plane. First, track-level features record information such as speed and location. Second, context features capture the interactions between tracks and computed scene contexts both on image and world coordinates. Third, composite features roughly categorize tracks to be human or vehicle based on heuristics using information such as bounding box size, speed, and location (on sidewalk or road etc). This level of information should capture all salient aspects of functional object behavior. The entire set of features are enlisted in



**Fig. 4.** (a-b) Two cluster results based on our approach (among total eleven). Each class delivers interpretable behaviors : parking and people on sidewalk. (c) A cluster obtained by [6]. Green and red marks indicate track heads and tails.

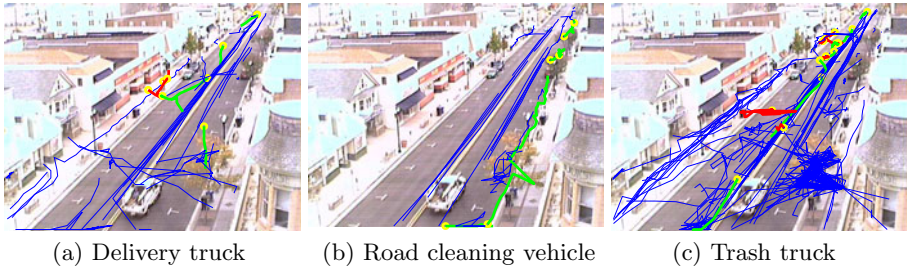
Table [1]. Note that Boolean features are not mutually exclusive, e.g., both composite features can be true, indicating that a track maybe both human and a vehicle, embracing the uncertainty and mitigating more strict decision to future computational modules. Context features capture interactions based on the changes of distance within every track to scene contexts. Fig. [3](c) shows a pre-computed context distance maps on image for the trash can scene context (shown in blue in Fig. [3](a)). Using distance maps on both image and world coordinates in conjunction with tracking, we can compute interactions efficiently. While most types of context features include 5 features, one each for every manual scene context, `Move_on` type includes 10 additional features, totaling 15. Tracks frequently move across different scene contexts, and 10 additional unordered pairwise fields were formed (out of 5), e.g., `Move_on_Sidewalk_and_ParkingLot`, to encode such behavior. Note that different subsets of features are used at different future processing modules (marked in Table [1]) which include link classifiers (L), elementary function clustering (C), and pruning function (P).

## 5 Functional Object Model Learning

To model functional object behavior over time in the presence of track fragmentations, we have formulated a two-level hierarchical approach. At the first lower-level, a codebook for individual track quantization is learned to provide a vocabulary of low-level activities based on encoded features. The learned codebook is then used to assign every track to one of different elementary functions. At the higher level, full function models are learned from a sequence of quantized tracks, regardless of the detailed feature information encoded in every track.

**Elementary Functional Model Learning.** Identification of elementary functional models is conducted by clustering all tracks based on contextual and composite features. We have explored four different clustering methods: K-means, mean-shift, spectral clustering [13], and affinity propagation [14]. The resulting clusters represent a group of tracks that share a common set of features. We have found that affinity propagation [14] produces superior clusters in terms of





**Fig. 5.** Three examples of functional objects: (a) delivery truck, (b) road cleaning vehicle, and (c) trash truck. Tracks belonging to human and vehicle movers are shown in red and green respectively where yellow circles mark the beginnings of tracks belonging to functional objects. Blue tracks are concurrent movers.

interpretability with minimal parameter tuning efforts. Fig. 4(a-b) show tracks within two sample clusters among total eleven, which deliver interpretations of vehicles parking and people on sidewalks, all independent of event locations. Although omitted for brevity, other clusters captured activities such as vehicles passing through and people crossing roads. As a comparison, a trajectory analysis method [6] has been applied where a sample cluster dominated by spatial distribution is shown in Fig. 4(c), only showing pedestrians on the left sidewalk.

**Full functional object model learning.** We adopted generative learning approaches for full functional model learning. This is because the size of positive training examples are often limited and the identification of negative examples are challenging for content-based retrieval problems. Our goal is to learn full functional model for long-term object activities from a given set of positive example which are assumed to be in the form of a sequence of manually linked tracks. For example, four functional objects which include postman, delivery truck, road cleaning vehicle and trash truck are shown in Fig. 11(b) and Fig. 5(a-c). It can be seen that each example consists of fragmented tracks. For delivery truck and trash truck, tracks even switch between multiple movers, i.e., people (red) and vehicle (green). Three modeling regimes are explored : unigrams, bigrams, and HMMs. Unigram simply counts unique symbols individually, while bigrams count the unique consecutive pairs. These three models provide different spectrum on amount of information they can capture. Since we use elementary functional models to discretize the tracks within sequences, every sequence example is represented as a series of symbols. For example, a sequence may be represented as: 1121 for a four-track sequence. Additionally, we insert gap variables, e.g., zero(0), whenever there is a temporal gap between tracks. This modification yields a sequence representation of : 1010201 (assuming temporal gaps everywhere). The number of parameters to be estimated for each of the three modeling regimes is in increasing order of unigrams, bigrams, and HMMs.

We use unigrams and bigrams within sample-based learning framework. This approach effectively converges to nearest-neighbor classifier. There are several well-known distance metrics that measure how distinct two bag-of-words distri-



**Fig. 6.** (a) Boosting algorithm decreases linkage classification errors as increasing number of weak classifiers are learned. The x-axis shows the progress of boosting learning as the number of weak classifiers, and y-axis shows classification error. Detected links overlaid (in cyan) on sequences of (b) a postman and (c) a delivery truck.

butions are. In this work, we used Bhattacharyya distance. For HMM learning, we used standard initialization with uniform priors and EM learning with Dirichlet priors to compensate for limited amount of training examples. The number of hidden states was set to be the number of elementary functions.

## 6 Track Linking, Pruning, and Linked Hypotheses

To recognize functional objects in the presence of track fragmentations, links between tracks should be identified to generate linked hypotheses. Our insight is that, if two fragmented tracks are a source and destination pair which belong to an identical mover, then a large portion of the feature distributions on these tracks will agree. Every track is encoded with comprehensive set of features (see Table I) which consist of two types of features : Boolean and continuous. If both Boolean features in two separate candidate tracks agree, we assign 'True' to the corresponding dimension in the new pairwise feature, and 'False' otherwise. For continuous features such as velocity and location, the absolute difference between the two values is computed. Additionally, we have included additional features which may be potentially useful: the number of total agreed Boolean features, and the distance between two tracks.

We have used 'Adaboost.M1' [1] with decision stumps to learn a linking function. Fig. 6(a) shows decreasing errors w.r.t. the increasing number of weak classifiers. Qualitative results of automatic linking on datasets of a postman and a delivery truck are shown in Fig. 6(b) & (c). (see Fig. 1(b) and Fig. 5(a) for originals) where the detected links (cyan) are overlaid. The tracks belonging to the primary function tracks are well-linked, with minor number of confuser links. Based on the outputs from Adaboost link classifier, track sequences with higher link probabilities are formed into functional behavior hypotheses to be evaluated against full functional models. The foremost concern with non-exclusive linking approach is that the number of hypotheses can grow (approximately) exponentially w.r.t. the identified potential links. For example, during our studies, we have seen impractically large number of 1 billion hypotheses are generated from 15 minute query video. It is crucial that number of links to be reduced. There

are two potential approaches to address this problem: (1) build object-specific linking function, and (2) prune tracks that seemingly do not belong to the functional object class of interest prior to linking. The first approach, however, did not work well because the limited number of training examples per class (often the case for content-based retrieval) made successful learning difficult.

Accordingly, a pruning scheme has been developed to filter out large portion of unrelated concurrent tracks prior to recognition, which leads to reduced number of hypotheses and alleviates computational demand substantially. Our approach is to prune out tracks which demonstrate little similarity to the positive example tracks belonging to the function class of interest. Our successful solution is to directly use the available Boolean context features on a track to measure the similarity against the provided training tracks (see Table I). When there are substantial number of identical Boolean context features between a candidate track and tracks within positive training dataset, we assume that it is more likely to be kept as candidate tracks, otherwise, it will be pruned out prior to linking. Similar to linking, we have used the number of agreed Boolean features as similarity score for pruning. To obtain a threshold  $\theta$  for pruning, min-max similarity across positive examples is used:  $\theta = \min(\{\theta_i | \theta_i = \max_{i \neq j}(\{\theta_{ij}\})\})$ . Here,  $\theta_{ij}$  denotes the similarity between two training tracks. For a novel tracklet, if there exists a training track with similarity score higher than the threshold, it is kept, otherwise, it will be pruned. In our test, the pruning module eliminated about 90% of negative examples while it kept most of the promising tracks (>97%). We have also explored the use of related max-min thresholds, however, it turned out to keep unnecessarily large number of tracks, lowering negative example pruning rate close to 50%. In summary, as the result of combined prior-pruning-then-linking approach, the number of generated hypotheses was reduced by several orders of magnitude where the maximum from a set of 15 minute videos was at most 2500, which is within manageable bounds.

## 7 Experimental Results

**Link Classifier and Hypotheses Generation.** To assess quantitative performance of the developed linking and hypotheses generation framework (Sec. 6), we have tested our work along with two additional linking methods on tracks collected from webcam data. First, we look into the generic linking accuracy of developed linking functions. By generic linking, we mean that linker accuracy will be assessed based on test dataset not being limited to the ones that contain functional object sequences. Linker function outputs either link probabilities (the case of Adaboost in our work) or link scores. Each of the  $i$ -th element in training data for AdaBoost is in the form of  $(x_i, y_i)$  where  $x_i$  is a multi-dimensional feature vector and  $y_i \in \{0, 1\}$  is a label. For AdaBoost training,  $N_p (\approx 250)$  positive examples and  $N_n (\approx 500)$  negative examples were used. As competing methods, we have implemented two additional linker functions and learned the parameters through training. The first link function is learned based on RankBoost [10]. The training data consists of pair of feature vectors  $(x_{i,0}, x_{i,1})$  where the preference/rank of the the first item is higher than the second. All the features used for

AdaBoost were re-used and additional track smoothness features which measure the kinematic continuity between tracks used in [10] were included. RankBoost learning process generates a strong rank function  $H(x) = \sum_t \alpha_t h_t(x)$  which consists of linear chain of weak ranker  $h_t(x)$  where we used decision stumps. It is desirable to assess the learning capability of AdaBoost and RankBoost given equal amount of training data. From the training data used for AdaBoost, we created  $(N_p + N_n)$  number of preference pairs where  $x_{i,0}$  and  $x_{i,1}$  belong to positive and negative training dataset respectively. In addition, as pointed out by [10], the outputs of RankBoost classifier does not deliver precise interpretations as probability within range [0,1]. We used logistic regression to map the outputs of RankBoost to probabilities. The second link function implemented is based on more traditional idea which outputs several costs  $\{c_i\}$  based on kinematic continuity and appearance similarity, e.g., [9]. In our work, we have considered such costs between tracks as link features and learned a logistic regression function  $1/(1 + e^{-\sum w_i c_i})$  as a link classifier where the weight parameters  $\{w_i\}$  were learned from available training data. Newly developed contextual features were not used for this linker for comparison purposes.

The generic-linking test results of all three link classifiers on test dataset of ( $\approx 300$ ) positive and ( $\approx 500$ ) negative examples are shown on the left side of Table. 2. Probability of detection (PD) and false positive rate (FP) are shown. A standard threshold of 0.5 was used as decision boundary. It can be observed that boosting-based methods outperform the traditional weighted score method in terms of PD while FPs are all similarly low. Boosting-based methods effectively exploit diverse features. On the other hand, the conventional features used for weighted score linker are presumably less reliable, mostly due to the low resolution and low frequency characteristics of the video clips. For example, low-level kinematics features which rely heavily on accurate high-frequency dynamic information is not captured well and often rejects true links by mistake. The superiority of AdaBoost over RankBoost can be attributed from two aspects. Theoretically, there is no guarantee that RankBoost will actually outperform AdaBoost for classification tasks. Accordingly, use of larger training dataset may be needed for superior performance.

In addition, we conducted experiments to assess the benefits of various sub-modules for identifying long-duration linked trajectories of functional objects, using 13 video clips containing functional objects. We applied all three linker functions along with two optional modules: pruning (P) prior to linking and Hungarian method (H) to impose exclusive links. The results of three combinations of approaches are reported on the right side of Table. 2. The average number of tracks per video clip was 103.9 where the average number of generated hypotheses and the PD which denotes the ratio of datasets where the generated hypotheses included full trajectories of functional objects are shown in Table. 2. If both optional modules are turned off, the space of all linked hypotheses becomes impractically large occasionally and the results are omitted. Two important observations can be made from Table. 2. First, the pruning process indeed reduces the number of generated linked hypotheses, nonetheless, it

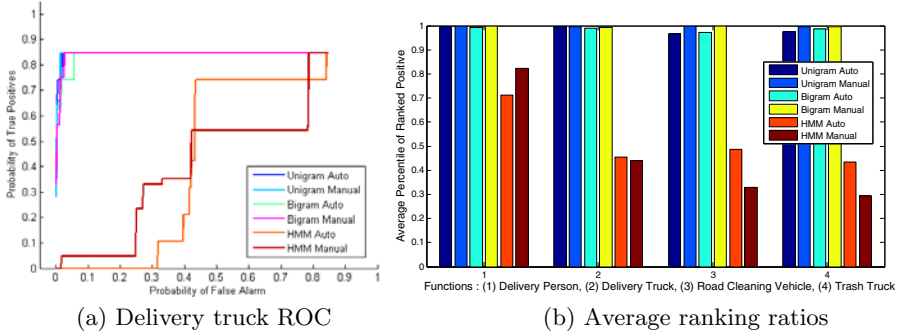
**Table 2.** Average performance statistics of different linking algorithms over total 13 datasets that contain functional objects. 'P' denotes the use of pruning step prior to linking. 'H' refers to the use of Hungarian method to impose exclusive linking.

	Generic Linking		Linked Hypotheses				
	PD	FP	#Tracks	P	H	# Hypotheses	PD
AdaBoost (ours)	0.82	0.08	103.9	○		341.8	0.85
					○	41.1	0.38
				○	○	35.2	0.38
				○		180.1	0.30
RankBoost (logistic regression)	0.70	0.05			○	29.3	0.08
				○	○	24.8	0.08
				○		130.2	0.08
Weighted scores (logistic regression)	0.61	0.09			○	24.4	0.0
			○	○	19.7	0.0	
			○				

does not affect the overall PD to be lower. Second, although exclusive linking substantially lower the number of generated hypotheses, PDs lower as well. The results suggest that, if considerations can make brute-force style search more affordable, it will be worthwhile to pursue such direction.

**Functional Object Recognition.** Our primary metric for functional object recognition is ROC curves. In our efforts, samples refer to the set of all generated hypotheses. If a sample contains the entire sequence of ground truth tracks, it is considered to be a 'hit', i.e., a super-set hypothesis of a positive sample is still a hit. Test samples are either scored by Bhattacharyya distances (for unigrams and bigrams), or by data likelihood (for HMMs). In terms of decision thresholds, we adopt top ranking number as such measure. Top ranking number denotes the number of highest-scored hypotheses that are to be returned as retrieval results.

Four functional objects selected from our webcam dataset were considered: postman, delivery truck, road cleaning vehicle, and trash truck. We have manually cropped total of 13 video clips from our webcam dataset where durations range from 3 to 15 minutes. Each of these clips loosely contains an example object along with approximately 1 min of extra amount of video before and after occurrences of examples. The original number of video clips for four functional classes are : 2-3-4-4. To obtain additional realistic data, we perturbed the obtained tracks to create three additional perturbed clones per track, resulting in dataset sizes of : 8-12-16-16. For the learning of elementary functional models and linking functions, we used original clips only (total 13). The exclusion of perturbed dataset during training was to assess the generalization power across different datasets. For every learning and recognition of full functional object trajectories, leave-one-out experiments are conducted. In the case of postman class, 7 positive examples are used in training process and one left-out example is included in the test dataset along with all the other available datasets which do not belong to the trained functional category (total 44). On average, the size of the target functional object class data constitutes about 3% of each batch of test sets. The ROC curve for a particular functional class is computed as the average



**Fig. 7.** (a) ROC curve for delivery truck dataset. 'Manual' and 'Auto' refer to the use of manual and learned scene context clusters respectively. Results of six different approaches: [Unigram, Bigram, HMM]  $\times$  [Auto, Manual]. Note : bigram results are similar to unigram results, and may not be visible due to unigram curves. (b) Average ranking ratios of for four functional objects under six varied experimental settings. Higher ranking ratios represent more accurate classification.

from multiple experiments conducted for that particular class. In particular, we have conducted the whole pipeline of learning and recognition experiments using both manual (see Fig 3(a)) and automatically obtained scene contexts (see Fig. 3(b)) to assess how much impact the accurate manual identification of scene contexts make. Then, normalized average rankings (AR) were obtained from the ranking results w.r.t. the total number of generated hypotheses. Accordingly, an AR close to 1.0 indicate accurate classification.

An example ROC curve for delivery truck class is shown in Fig. 7(a). An interesting outcome is the promising performance of unigrams and bigrams: both achieve very high PDs at very low cost of FPs. Considering the simplicity and computational efficiency associated with these models, the accurate identification results show that they can capture the general characteristics of particular functional classes well enough to yield favorable recognition. Another finding from the ROC curve analysis is that simpler models such as unigrams and bigrams are outperforming more sophisticated counterpart such as HMMs. The observed weak performance of HMMs can be explained from the generalization point of view. Given the limited number of training samples which ranges from 7 to 15 samples, comparably far larger number of HMM parameters fail to capture the general characteristics of data. These encouraging results for simpler models are likely due to the current problem setting: content-based learning with limited number of training examples. Analogous analysis can be drawn from the average ranking ratio results for all four functional objects under six experimental settings, shown in Fig. 7(b). The ROC curves for the other three object classes showed very similar characteristics (omitted for brevity). Note that among four classes, delivery truck and trash trucks are more challenging because they include both person and vehicle movers. No hit hypothesis could be generated from a few datasets belonging to these two classes, accordingly, corresponding

ROC curves never achieve perfect PD of 1.0 (see Fig. 7(a)). Additional finding is that the overall recognition degrades only marginally even when unsupervised scene contexts are used, in comparison to more accurate manual contexts. This observation suggests that rough identification of scene contexts may be sufficient to deliver favorable functional object recognition results in many cases. It would be worth noting that our results may be more optimistic than true reality, primarily due to the facts that perturbed data is used and the ratio of true positive examples may be less in practice than our current experimental setting, probably appearing less than 1% of time while these objects constitute average 3% of data in this work. We plan to investigate these issues in our future work.

## References

1. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and System Sciences* 55, 119–139 (1997)
2. Stark, L., Bowyer, K.: Achieving Generalized Object Recognition through Reasoning about Association of Function to Structure. *PAMI* 13, 1097–1104 (1991)
3. Peursum, P., West, G., Venkatesh, S.: Combining image regions and human activity for indirect object recognition in indoor wide-angle video. In: *ICCV* (2005)
4. Gupta, A., Davis, L.: Objects in Action: An Approach for Combining Action Understanding and Object Perception. In: *CVPR* (2007)
5. Junejo, I., Javed, O., Shah, M.: Multi feature path modeling for video surveillance. In: *ICPR* (2004)
6. Wang, X., Ma, K.T., Ng, G.W., Grimson, E.: Trajectory analysis and semantic region modeling using a nonparametric Bayesian model. In: *CVPR* (2008)
7. Yang, Y., Liu, J., Shah, M.: Video scene understanding using multi-scale analysis. In: *ICCV* (2009)
8. Chan, M., Hoogs, A., Schmiederer, J., Petersen, M.: Detecting rare events in video using semantic primitives with HMM. In: *CVPR* (2006)
9. Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W.: Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: *CVPR* (2006)
10. Li, Y., Huang, C., Nevatia, R.: Learning to associate: Hybridboosted multi-target tracker for crowded scene. In: *CVPR* (2009)
11. Turek, M.W., Hoogs, A., Collins, R.: Unsupervised learning of functional categories in video scenes. In: *ECCV* (2010)
12. Swears, E., Hoogs, A.: Functional scene element recognition for video scene analysis. In: *IEEE Workshop on Motion and Video Computing* (2009)
13. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: *NIPS* (2006)
14. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* 315, 972–976 (2007)

# Anomalous Behaviour Detection Using Spatiotemporal Oriented Energies, Subset Inclusion Histogram Comparison and Event-Driven Processing

Andrei Zaharescu<sup>1,2</sup> and Richard Wildes<sup>2</sup>

<sup>1</sup> Aimetis Corporation, Waterloo, Canada

<sup>2</sup> Department of Computer Science and Engineering,  
York University, Toronto, Canada  
{andreiz, wildes}@cse.yorku.ca

**Abstract.** This paper proposes a novel approach to anomalous behaviour detection in video. The approach is comprised of three key components. First, distributions of spatiotemporal oriented energy are used to model behaviour. This representation can capture a wide range of naturally occurring visual spacetime patterns and has not previously been applied to anomaly detection. Second, a novel method is proposed for comparing an automatically acquired model of normal behaviour with new observations. The method accounts for situations when only a subset of the model is present in the new observation, as when multiple activities are acceptable in a region yet only one is likely to be encountered at any given instant. Third, event driven processing is employed to automatically mark portions of the video stream that are most likely to contain deviations from the expected and thereby focus computational efforts. The approach has been implemented with real-time performance. Quantitative and qualitative empirical evaluation on a challenging set of natural image videos demonstrates the approach's superior performance relative to various alternatives.

## 1 Introduction

Detection of anomalous behaviour relative to some model of expected behaviour is a fundamental task in surveillance scenarios. Examples include detection of movement in an area where none should occur (as in a secure storage facility) and detection of “wrong way motion” where movement of objects only should occur in one direction yet are observed in a different direction (as in movement of traffic on a one-way road). In particular, given the increase in video coverage of public and private spaces, an automated ability to monitor the acquired data and signal deviations from expected behaviour would be very useful, as it could serve to alert either human or artificial systems to analyze further the data that is acquired.

A number of challenges must be surmounted for successful detection of anomalous behaviour in surveillance video. In essence, these challenges arise from the need to model a wide range of potentially complicated patterns of normal activity and detect fine deviations from that model, even while being robust to changes that are insignificant. Normal activity can range from simple no temporal change through single and



multiple motions to complicated situations of dynamic textures (e.g., backgrounds of fluttering vegetation or water waves), including multimodal behaviour. Modeling must be flexible to encompass this entire range. Anomaly detection must be able to register subtle changes of interest (e.g., changes in direction or speed of motion, presence of a coherently moving object against a camouflaging background of texture and dynamic clutter), while not signaling insignificant changes (e.g., naturally occurring illumination changes during the diurnal cycle, differences between objects that are manifest purely in terms of spatial appearance without behaviour differences). It also is desirable to allow for partial matches of observations to the model, as complicated scenarios might encompass multimodal behaviour and observations that correspond to any modeled mode are acceptable, while alternatives are not. Further, in many situations an ability to incorporate deviations that recur over time into the model is desirable, so that they are no longer considered anomalous.

**Related Work.** One general class of approach to anomaly detection in video is based on explicit tracking of viewed objects [22,31,9,18,5]. Such approaches acquire models of typical trajectories from tracker output over some training period and subsequently signal deviations in observed tracks as anomalies. A significant limitation to this class of approaches is their reliance on (visual) tracking, a still unsolved challenge.

Background subtraction techniques that model typical appearance from a camera view can be applied to detecting behaviour anomalies (see, [27] for review and, e.g., [14,20,36,17] for a sampling of more recent work). The simplest techniques involve unimodal background models of pixelwise image intensity and have limited applicability for complicated backgrounds. Increased sophistication in modeling static background appearance comes through consideration of pixel attributes beyond image intensity (e.g., gradients, edges, texture). More involved techniques account for dynamic backgrounds by acknowledging multimodal intensity distributions, parametric modeling, kernel-based estimation and predictive filtering. An extension of predominantly intensity-based background modeling for video operates by indexing observations relative to a database of normal videos, with failures taken as anomalies [8]. A limitation of appearance-based approaches is their inability to abstract purely dynamic aspects of behaviour, which can lead to overly restrictive, under-generalized models of normal behaviour (e.g. lack of invariance to different actors performing the same activity).

More closely related to the approach proposed in the current paper are efforts that have more explicitly modeled the dynamic behaviour of backgrounds. Typically, such approaches make use of some type of spatiotemporal filtering to define normal local activity with anomalies taken as deviations from the defined model. Along these lines, some work has appealed directly to spatiotemporal gradient measurements [30,25]. Other work has been more restricted to considering only the temporal first derivative (blurred and quantized) [35]. Alternatively, image flow measurements have been used to define local activity models [7,4,23,2,24]. Still other work has abstracted local flow measurements to a simpler consideration of whether or not a pixel typically is in motion to define locally normal behaviour [21]. Direct use of spatiotemporal gradients to define normal activity has a number of limitations, including sensitivity to image contrast and spatial pattern, which lead to lack of robustness to changes in illumination and

different appearing actors performing the same activity. Further, reliance on temporal derivative alone leads to an inability to distinguish different motion directions. Alternatively, approaches that rely on (local) flow measurements are limited in the complexity of behaviours they can capture, e.g., multiple motions at a point, temporal flicker and dynamic textures (e.g., water, wind-blown foliage) can be difficult to model, as they violate the underlying assumptions of the flow computation (e.g., brightness conservation) and thereby yield unreliable results in such scenarios.

A number of recent approaches are concerned with modeling of non-local behaviour (but typically building on local measurements) with application to anomaly detection [6,25,24,29,28,32,26]. While such approaches make strides in accounting for non-local activity, they still can be limited by overly restrictive local representations, e.g., spatiotemporal gradient models that are not invariant to spatial appearance and flow models that do not account for activity that is amenable to characterization as a single local flow vector (e.g., multiple motions and more general dynamic textures).

To account for complicated local behaviour, measures of spatiotemporal oriented energy play a prominent role in the approach proposed in the current paper. Previously, such measures have been used in a variety of vision processing tasks, including image enhancement and motion estimation [16], video segmentation [12], pattern categorization [34] and activity recognition (although not generic anomaly detection) [10,13,11].

**Contributions.** In the light of previous research, the present approach makes four main contributions. 1) 3D,  $(x, y, t)$ , spatiotemporal oriented energy measurements are used to represent observations. While almost any approach to anomalous behaviour detection must employ spatiotemporal filtering of some type, no previous work has made use of the energy filtering framework proposed here, which enjoys a number of benefits in being able to capture a wide range of image dynamics (both standard motion as well as more complex dynamic patterns, e.g., flickering lights, swaying vegetation and water), even while being robust to irrelevant variations (e.g., overall illumination variation and different appearing individuals engaged in the same behaviour). 2) A novel histogram comparison method is presented to detect anomalous behaviour relative to an acquired model. A key component of this measure is that it accounts for partial matches of new observations to the acquired model. 3) Event-driven processing is used to automatically mark portions of the video stream that are most likely to correspond to activities and thereby focus computational efforts. 4) The proposed approach has been realized in real-time implementations. A detailed empirical evaluation of the implementations is presented, which documents the contributions of its individual components and its strong overall performance relative to alternative approaches.

## 2 Technical Approach

The developed approach to detecting anomalous behaviour is based on observed deviations from an acquired model of normal behaviour. The model is image-based and thereby indicates expected (normal) observations on a pixelwise basis as recorded from a specific viewpoint.

### 2.1 Spatiotemporal Energy Representation

In the developed approach, both model and newly acquired video observations are represented in terms of local distributions of 3D,  $(x, y, t)$ , spatiotemporal oriented energy as derived from input imagery via application of an orientation tuned filter bank. This representation is selected as it captures the local first-order correlation structure of visual spacetime and thereby allows a wide range of dynamic activities to be captured (e.g., both single and multiple motions as well as more general dynamic textures) with robustness to illumination and purely spatial appearance [12]. In particular, the current approach to spatiotemporal orientation for anomaly detection follows closely the previous work [12], where it was used instead for video segmentation.

To extract the orientation measurements, oriented energy filtering is realized in terms of second derivative of 3D Gaussian filters,  $G_{2\theta}(x, y, t)$ , and their Hilbert transforms,  $H_{2\theta}(x, y, t)$ , where  $\theta$  represents the direction of the filter’s axis of symmetry. These particular filters are selected due to their (moderately) broad tuning, which allows for a wide range of orientations to be captured with a relatively small number of filters. Additionally, these filters admit a steerable and separable formulation [15], which leads to efficient computations. The filters are taken in quadrature, to yield the following local oriented energy measure,

$$E_{\theta}(x, y, t) = (G_{2\theta} * I)^2 + (H_{2\theta} * I)^2, \tag{1}$$

where  $I \equiv I(x, y, t)$  denotes the input imagery and  $*$  symbolizes convolution.

For the case of dynamic spacetime orientation (e.g., as related to motion phenomena), each of the oriented energy measurements, (1), is confounded with spatial orientation. Correspondingly, the same pattern of activity will yield different responses across an ensemble of oriented energy filters depending on variations in the spatial appearance of the viewed object/event: This is an undesirable state of affairs for dynamic anomaly detection as it would not be possible to build models of normal behaviour that are robust to irrelevant details of purely spatial appearance (e.g., sensitivity to what people are wearing, when the concern is for how they are moving). To remove this difficulty, the spatial orientation component of the oriented energy responses is discounted by marginalizing this attribute via pointwise, linear combination of energy measures, (1), that support a single spacetime orientation, as specified by the unit normal,  $\mathbf{n}$ , corresponding to its frequency domain plane. (Recall that a pattern exhibiting a single spacetime orientation, e.g., velocity, manifests as a plane through the origin in the frequency domain [33].) In particular, the energy measure, (1), is refined to become

$$\tilde{E}_{\mathbf{n}}(x, y, t) = \sum_{i=0}^N E_{\theta_i}(x, y, t), \tag{2}$$

where  $\theta_i$  represents one of  $N + 1$  equal spaced orientation tunings consistent with direction  $\mathbf{n}$  and  $N = 2$  is the order of the Gaussian derivative filter (1), for details see [12].

The resulting oriented energies are confounded with local contrast. This makes it impossible to determine whether a high response from a particular filter is indicative of a close match with the underlying structure or is instead a low match that yields a

high response due to significant contrast in the signal. To arrive at a purer measure of oriented spacetime structure, the energy measures are normalized by the sum of the oriented responses at each point,

$$\hat{E}_{\mathbf{n}_i}(x, y, t) = \frac{\tilde{E}_{\mathbf{n}_i}(x, y, t)}{\sum_{\mathbf{n}_j \in \mathcal{S}} \tilde{E}_{\mathbf{n}_j}(x, y, t) + \epsilon}, \quad (3)$$

where  $\mathcal{S}$  denotes the set of (marginalized) oriented energies, (2), with  $\mathbf{n}_j$  a particular sample and  $\epsilon$  a constant, set to 1% of the maximum filter response, introduced as both a noise floor and to avoid instabilities at points where the overall energy is small.

In the currently implemented representation,  $K = 6$  different directions,  $\mathbf{n}$ , are made explicit, that correspond to leftward, rightward, upward and downward motion (each with peak response at 1 pixel/frame movement), static (orientation orthogonal to the image plane) and flicker (orientation orthogonal to the temporal axis); although, due to the broad tuning of the filters employed, responses arise to a wide range of orientations about the peak tunings. By construction, these measures are marginalized for purely spatial appearance and normalized for contrast, which allows for a degree of robustness to unimportant variability in observations. Further, the representation is simply realized by an alternating series of linear (i.e., separable convolution and pointwise addition) and pointwise non-linear operations (i.e., squaring and division); thus, efficient computations are realized.

Finally, it is straightforward to extend the described approach to multiple scales. In particular, the input imagery is brought under a pyramid representation [19] prior to filtering. Subsequently, the oriented filtering, (1), appearance marginalization, (2), and normalization, (3), are performed separately at each pyramid level to realize a multi-scale oriented energy representation. In the current implementation  $\sigma = 5$  scales are employed, with factor of  $\sqrt{2}$  subsampling between levels and commensurate lowpass filtering prior to subsampling.

## 2.2 Model Acquisition and Maintenance

The proposed model is given in terms of a histogram of spatiotemporal orientations observed over some period of time. Since behaviours of interest are (by definition) dynamic, only measures of orientation that arise from non-static observations are explicitly represented in the model. In particular, a key component to the method is the concept of accumulating statistics only on interesting events: The information is aggregated at the pixel level only between frames containing dynamic energy. Dynamic energy is captured in terms of a threshold,  $\beta$ , on the static channel  $E_{Static}$ : If static energy is greater than  $\beta$ , it is considered that there is no activity at the current pixel. To formalize the notion of event-driven processing, let

$$\psi(x, y, t) = \begin{cases} 1 & \text{if } E_{Static} < \beta \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

and the model histogram,  $\mathbf{m}(x, y)$ , be defined as

$$m_{\mathbf{n}}(x, y) = C \sum_{t=1}^{t=T} \psi(x, y, t) \hat{E}_{\mathbf{n}}(x, y, t) \quad (5)$$

where  $m_{\mathbf{n}}(x, y)$  is the histogram bin corresponding to orientation  $\mathbf{n}$  at location  $(x, y)$ ,  $C$  is a normalization factor ensuring the histogram sums to unity and  $t$  indexes from an initial to frame  $T$  used in building the model. The histogram at a given spatial location over a period of time is built by concatenating the relative energy of the  $K$  spacetime orientations,  $\mathbf{n}$ , at each of the  $\sigma$  scales, thus leading to a  $K \times \sigma$  bin histogram.

Similarly, a new observation is made by constructing a histogram,  $\mathbf{o}(x, y)$ , analogous to the model, except that it is accumulated only over a relatively small number of frames. In particular,

$$o_{\mathbf{n}}(x, y) = C \sum_{t=t_0-\lfloor(k/2)\rfloor}^{t_0+\lfloor(k/2)\rfloor} \psi(x, y, t) \hat{E}_{\mathbf{n}}(x, y, t) \tag{6}$$

where  $o_{\mathbf{n}}(x, y)$  is the histogram bin corresponding to spatiotemporal orientation  $\mathbf{n}$  at location  $(x, y)$ ,  $C$  is a normalization factor ensuring the histogram sums to unity and  $t$  indexes across  $k$  frames,  $k \ll T$ , that are used in accumulating the current observation at time  $t = t_0$ .

Finally, the model  $\mathbf{m}^t(x, y)$  at time  $t$  is updated in an ongoing fashion so as to account for the current observation,  $\mathbf{o}^t(x, y)$ , according to

$$\mathbf{m}^{t+1}(x, y) = [1 - \delta\psi(x, y, t)]\mathbf{m}^t(x, y) + \delta\psi(x, y, t)\mathbf{o}^t(x, y) \tag{7}$$

on a bin-by-bin basis with  $\delta$  controlling the update rate. Notice that update is only performed when there is an event  $\psi(x, y, t)$ , (4).

### 2.3 Comparison of Model and New Observations

Given a model,  $\mathbf{m}(x, y)$ , and a current observation,  $\mathbf{o}(x, y)$ , anomalous behaviour is defined in terms of deviations of the observation from the model. Given that both the model and observation are captured as histograms, various standard comparison methods might be invoked (e.g.,  $\chi^2$  test of independence or Bhattacharyya coefficient). However, such standard methods fail to capture two key points of relevance for anomaly detection. First, the observation might only encompass a subset of modeled activity: This easily can be the case, due to the fact that the model statistics typically are accumulated over a relatively large number of frames, possibly incorporating multiple activities (e.g., left and right motions); whereas, the observation statistics capture relatively shorter time periods that might not encompass all modeled activities (e.g., left motion only). Second, it is desirable to model scenarios where a lack of activity in the current observation histogram should not be considered anomalous, even if the previously acquired model for that particular pixel differs significantly.

To address the noted points, the  $\chi^2$  test of independence (3) is taken as a point of departure and modified, as follows. In the current context, the  $\chi^2$  measure between  $\mathbf{m}(x, y)$  and  $\mathbf{o}(x, y)$  is given as

$$\chi^2[\mathbf{m}(x, y), \mathbf{o}(x, y)] = \sum_{\mathbf{n} \in \mathcal{S}} \frac{(m_{\mathbf{n}}(x, y) - o_{\mathbf{n}}(x, y))^2}{m_{\mathbf{n}}(x, y) + o_{\mathbf{n}}(x, y)}. \tag{8}$$

The first point, regarding any particular current observation not encompassing all possibilities captured in the model, can be addressed by introducing a notion of subset

inclusion, i.e., the observed behaviour must be a subset of the modeled behaviour; else, it will be taken as anomalous. To indicate such anomalies, a function is needed that selects particular orientations (histogram bins),  $\mathbf{n}$ , where there is little response in the model (e.g., relative to some threshold,  $\tau_0$ ) even while there is significant response in the observation (e.g., relative to some threshold,  $\tau_1$ ); a corresponding function can be defined as

$$\phi(m_{\mathbf{n}}, o_{\mathbf{n}}) = \begin{cases} 1, & \text{if } (m_{\mathbf{n}} < \tau_0) \text{ and } (o_{\mathbf{n}} - m_{\mathbf{n}} > \tau_1) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

The second point, regarding a particular observation not encompassing any activity, can be addressed by assigning decreasing weight to an observation as fewer of the frames observed in its construction drive event-based processing as indicated by (4). This notion can be captured by an event ratio,  $\rho[\mathbf{o}(x, y)]$ , of the number of frames that contributed to the event-based processing,  $\gamma[\mathbf{o}(x, y)]$ , to the total number of frames observed,  $\alpha[\mathbf{o}(x, y)]$ , i.e.,

$$\rho[\mathbf{o}(x, y)] = \frac{\gamma[\mathbf{o}(x, y)]}{\alpha[\mathbf{o}(x, y)]}. \quad (10)$$

Combining the original  $\chi^2$  formulation, (8), with the formalization of subset inclusion, (9), and event ratio, (10) yields the final measure of distance between a model and observation

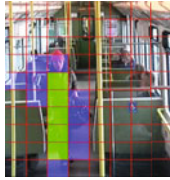
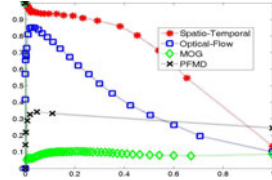
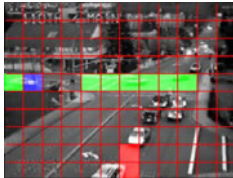
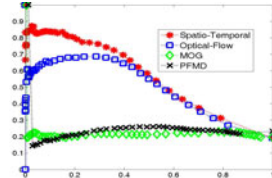

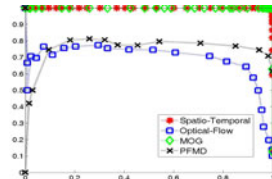
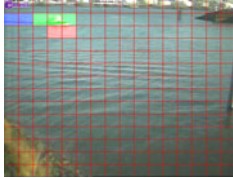
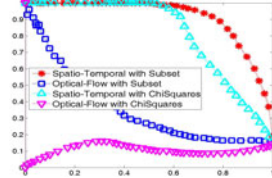

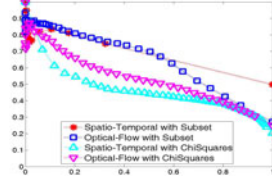
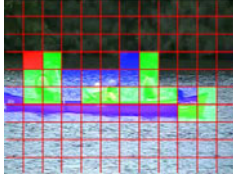
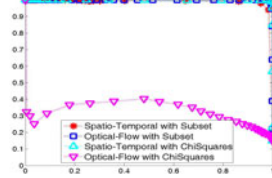
$$D[\mathbf{m}(x, y), \mathbf{o}(x, y)] = \rho(\mathbf{o}) \sum_{\mathbf{n} \in \mathcal{S}} \phi(m_{\mathbf{n}}, o_{\mathbf{n}}) \frac{(m_{\mathbf{n}} - o_{\mathbf{n}})^2}{m_{\mathbf{n}} + o_{\mathbf{n}}}, \quad (11)$$

with larger distances taken as increased evidence for behaviour anomaly at  $(x, y)$  and final anomaly detection based on a comparison to a threshold,  $\Delta$ . (Explicit reference to image coordinates,  $(x, y)$ , is suppressed on the right-hand side of the final distance measure, (11), for the sake of notational compactness.)

### 3 Empirical Evaluation

Three implementations of the proposed approach to detecting anomalous behaviour have been developed, which differ according to their software and hardware utilization and are documented in Table 1. Algorithmic parameters are the same for all implementations:  $\beta = 0.35, \delta = 0.005, \tau_0 = 1.5/h, \tau_1 = 0.15/h$ , where  $h$  is the number of histogram bins, i.e.,  $h = 6$  (orientations)  $\times 5$  (scales) = 30, unless otherwise noted. The reported timings are with respect to processing an image of size  $160 \times 120$  and attest to the applicability of the approach to real world operational scenarios. Detection results reported below are with respect to the naive ANSI C implementation; although, all implementations yield similar results.

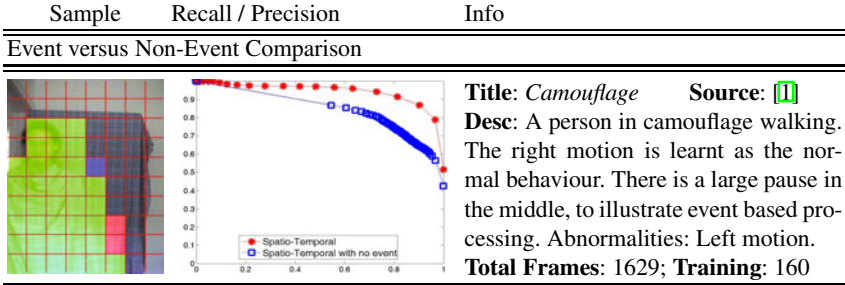
The implementations have been evaluated on a test suite of video sequences, which are documented in Figs. 1 and 2; actual videos are provided in the supplemental material. All sequences are of spatial dimensions  $320 \times 240$ . Each sequence was manually groundtruthed for anomalous behaviours relative to the depicted backgrounds. For the sake of practicality, images were groundtruthed on a coarse spatial grid of cells, shown

Sample	Recall / Precision	Info
<b>Representation Comparison</b>		
		<p><b>Title:</b> <i>Train</i>      <b>Source:</b> [1]</p> <p><b>Desc:</b> Very challenging train sequence due to drastically varying lighting conditions and camera jitter. Abnormalities: People movement.</p> <p><b>Total Frames:</b> 19218; <b>Training:</b> 800</p>
		<p><b>Title:</b> <i>Bellevue</i>      <b>Source:</b> [1]</p> <p><b>Desc:</b> Cars moving through an intersection. Model construction during day; testing continuing through night. Abnormalities: Cars entering thoroughfare from left or right.</p> <p><b>Total Frames:</b> 2918; <b>Training:</b> 200</p>
		<p><b>Title:</b> <i>Boat-Sea</i>      <b>Source:</b> [1]</p> <p><b>Desc:</b> A sea-boat is passing by (motion on motion). Abnormalities: Boat movement.</p> <p><b>Total Frames:</b> 450; <b>Training:</b> 200</p>
<b>Subset Inclusion versus <math>\chi^2</math> Histogram Comparison</b>		
		<p><b>Title:</b> <i>Boat-River</i>      <b>Source:</b> [1]</p> <p><b>Desc:</b> Boat passing by on a river (motion on motion). Abnormalities: Boat movement.</p> <p><b>Total Frames:</b> 250; <b>Training:</b> 80</p>
		<p><b>Title:</b> <i>Subway-Exit</i>      <b>Source:</b> [2]</p> <p><b>Desc:</b> Surveillance camera observing pedestrians at a subway exit. Abnormalities: Wrong way motion (leftward and downward).</p> <p><b>Total Frames:</b> 32426; <b>Training:</b> 6900</p>
		<p><b>Title:</b> <i>Canoe</i>      <b>Source:</b> [21]</p> <p><b>Desc:</b> A canoe is passing by (motion on motion); also, some wind-blown foliage in background. Abnormalities: Canoe movement.</p> <p><b>Total Frames:</b> 1050; <b>Training:</b> 200</p>

**Fig. 1.** The first column shows a frame during the evaluation of the proposed method, using the manually marked groundtruth information. The Colour coding is: green - true positive; red - false positive; blue - false negative. The second column presents the Precision/Recall curves (abscissa- Recall; ordinate - Precision), with each curve containing 20 measurements. The last column provides additional documentation for each example.

**Table 1.** Implemented instantiations of the approach for anomalous behaviour detection

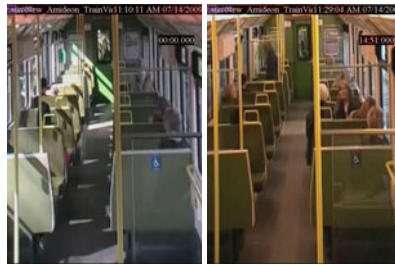
Language Device		Clock Cores	Time
ANSI C	Intel Core2Duo	2.4GHz	1 80 ms
SSE2	Intel Core2Duo	2.4GHz	1 24 ms
OpenCL	NVIDIA 280GTX	1GHz	120 5 ms

**Fig. 2.** Same formatting as in Figure 1

overlaid on the images. All the videos, the groundtruth data, as well as the groundtruth and the evaluation software are available online [1]. Quantitative evaluation is presented in the form of Precision-Recall (PR) curves by varying the detection threshold,  $\Delta$ , on (11), where  $Recall = \frac{\# \text{ True Positives}}{\# \text{ Positives in Dataset}}$  and  $Precision = \frac{\# \text{ True Positives}}{\# \text{ True Positives} + \# \text{ False Positives}}$ . In calculating the PR curves, false positive/negative cells adjacent to a true positive cell are discarded.

As detailed in Section 2, the proposed approach to anomaly detection centres around three key ideas: (i) behaviour modeling in terms of a distribution (histogram), (5), of spatiotemporal oriented energy responses, (3), (ii) model and observation comparison via subset inclusion, (11), and (iii) event-based processing, (4). The experiments document how each of these components contribute to the success of the proposed approach.

**Experiment 1.** The benefits of representation via a **distribution of spatiotemporal oriented energies** are manifested in cases that require robustness to variable illumination and camouflage, even while making fine distinctions between normal and abnormal ac-

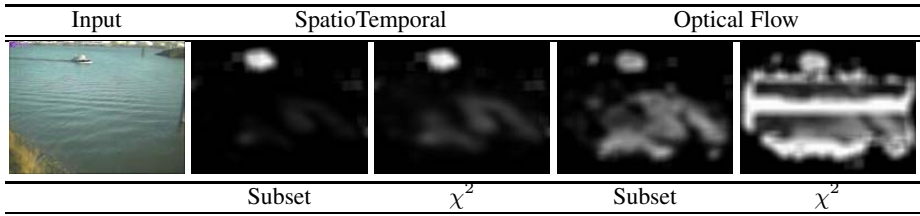
**Fig. 3.** Example images from *Train* sequence. Extreme background changes are present, as the moving train passes through highly variable exterior lighting conditions.



tivity. A striking example of variable illumination is presented in *Train*, which includes sudden, extreme background changes caused by the moving train passing through tunnels, see Fig. 3 for two dissimilar backgrounds taken shortly apart and the supplemental video. As discussed in Sec. 2.1, the bandpass nature, (1), and response normalization, (3), of the employed filtering make the representation invariant to additive and multiplicative intensity changes and these properties yield the strong performance in variable illumination shown in Fig. 1. Robustness of the proposed approach to more gradual changes in illumination is illustrated in *Bellevue*, as the sequence begins during day and progresses through night. Also of interest in this case is clutter caused by headlights with the onset of dusk.

Spatial camouflage, where novel objects have the same texture patterns as their surround also are not problematic for the proposed approach, as the representation emphasizes distinctions on the basis of dynamics; an example is shown in *Camouflage* where the moving person is covered with the same spatial texture pattern as the background. Dynamic camouflage can come about when normal behaviour is sufficiently erratic to mask novel movement. Representation in terms of a distribution of spatiotemporal orientations allows for such camouflage to be broken, as a wide range of image dynamics can be captured and distinguished: The approach can encompass complicated background dynamics in its model (e.g., motion jitter and rapidly moving shadows/lights in *Train*, and variable waves in *Boat-Sea* and *Canoe*), yet still detect novel moving objects as anomalies (e.g., people, boat and canoe in *Train*, *Boat-Sea* and *Canoe*, resp.). Similarly, since different directions of motion can be distinguished, an observed set of motion directions can be incorporated into the model, while alternative motion directions are marked as anomalous (e.g., wrong-way motion detection of *Bellevue*, *Subway* and *Camouflage*).

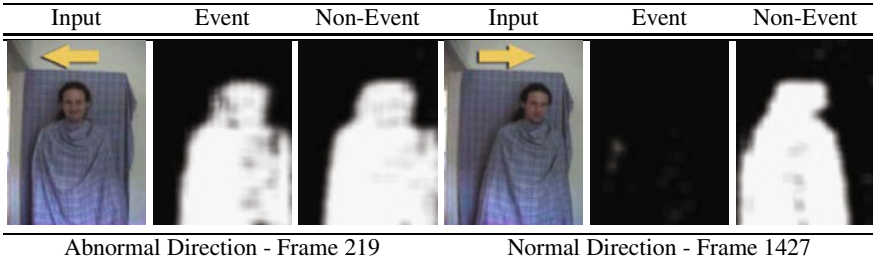
The benefits of the proposed representation are quantified by the PR curves for *Train*, *Bellevue* and *Boat-Sea* in Fig. 1 where a comparison is made to three alternatives. The first is image intensity-based: Capturing behaviour via pixelwise image intensity Mixture of Gaussians (MOG) [27], with a MOG model of normal behaviour acquired during a training period and subsequent intensity observations judged as anomalies based on the joint posterior probability that they belong to any of the modeled modes. The second alternative representation is motion-based: Capturing behaviour via pixelwise Percentage of Frames Motion is Detected (PFMD) [21], with motion detection performed using the opponent spatiotemporal energy magnitude ( $|E_{up} - E_{down}|^2 + |E_{left} - E_{right}|^2 > 0.05$ ), which in preliminary experiments yielded superior performance to temporal differencing used elsewhere for PFMD modeling [21]. (Notice that opponent spatiotemporal energy magnitude will be relatively large in response to a locally coherent motion [34].) Both of these representations were embedded in the recently proposed *behaviour subtraction* method of anomaly detection [21], as it readily handles both MOG and PFMD models; whereas, the method proposed in the present paper is more specialized for distributed (histogrammed) measurements. The third alternative is based on quantized optical-flow direction and magnitude computed at multiple, 5, scales (e.g., as originally proposed for direction or magnitude [2] and subsequently extended to combine 8 directions with magnitude [24]; the latter is used here, as it was found to provide superior performance in preliminary experimentation). The quantized optical flow



**Fig. 4.** Comparison of proposed (subset inclusion), (11), vs.  $\chi^2$ , (8), histogram comparison measures for the Boat-River sequence (frame 161)

defines a histogram [224] that is substituted directly into the proposed approach by substituting for oriented energy; thus, a direct comparison is had between oriented energy and optical flow, as all other system components are constant. With one exception, it is seen that the alternative representations yield notably lower PR curves in comparison to the proposed approach, as they are not able to encompass the complicated normal behaviour that is present in the examples. The sole exception is the case of MOG applied to *Boat-Sea* where the appearance of boats (abnormal) are sufficiently different from the acquired mixture that performance is comparable to the spatiotemporal representation. Still, optical flow appears to be second best for the other two cases, *Train* and *Bellevue*.

**Experiment 2.** The main benefit of comparing model, (5), and observation, (6), histograms via **subset inclusion**, (11), is that it allows for partial fits between observations and models. This property is important so that every given observation does not need to encompass the entire range of previously modeled behaviour. To illustrate the practical importance of this consideration, Fig. 4 shows comparative image results of subset-inclusion vs.  $\chi^2$  histogram comparison (all other components are exactly the same as those of the proposed method); associated PR curves are shown in Fig. 11. Here, PR curves are shown for both spatiotemporal oriented energy as well as optical flow, as quantized flow can be substituted directly for the energies in the proposed approach (see Exp. 1) to show the benefits of subset inclusion beyond application to energy measurements. Also, flow appeared to be the second best overall performer when comparing representations in Exp. 1. For *Boat-River* and *Subway* using energy as well as flow, it is seen that for a given recall rate,  $\chi^2$  has a strong tendency for lower precision relative to subset-inclusion. For *Canoe* spatiotemporal energy already is performing extremely well with just  $\chi^2$ ; however, addition of subset-inclusion allows flow to elevate its level of performance to that of energy. These results are readily explained as  $\chi^2$  is not able to accept as normal partial matches to the model; whereas, subset inclusion is with resulting higher precision in its detection, i.e., fewer false positives. The quantitative summaries are supported in the pictorial results, especially for complicated backgrounds (e.g., water in *Boat-River* and water/vegetation in *Canoe*, which encompass a range of motions; whereas, any particular observations show only a subset and such partial matches are reported as anomalies by  $\chi^2$ , but not by subset-inclusion. Finally, notice that flow leads to similar performance to spatiotemporal oriented energy on *Subway*. This can be accounted for by the fact that both normal and abnormal behaviours



**Fig. 5.** Comparison of event vs. non-event based update schemes. Without event-based processing, the normal behaviour (right motion) is forgotten after 300 frames of no activity (starting at frame 803) and it is incorrectly detected as abnormal. Event-based processing successfully maintains the model and it does not yield false positives.

(motion of pedestrians) can be captured well by flow (as well as by spatiotemporal oriented energy). Just in this example alone, 10 orientations have been used for spatiotemporal energies by adding 4 directions aligned with motion along diagonals (e.g., up-left, up-right, etc.) to the standard set of 6 (only 4 of which are aligned with motion directions, left, right, up, down), in order to bring its directional discrimination more on par with the optical flow representation, which explicitly encodes motion along diagonals in its histogram bins (as well as left, right, up and down, plus magnitude). Using only 6 orientations for spatiotemporal energy in this example led to performance slightly worse than flow in preliminary experiments, owing to poorer (motion) direction resolution.

**Experiment 3. Event-based processing** influences construction of models, (5), (7), and observations, (6), to focus computations on portions of the data where behaviour is occurring, as signaled by events, (4). Not only does such processing reduce computational load (e.g., fewer updates are performed), but it also keeps models and observations defined in terms of dynamic behaviour. An interesting benefit of this processing is that it ameliorates problems with forgetting aspects of normal behaviour during model update: Without event-based processing, a modeled event will be discarded from the current model after  $1/\delta$  frames by the update, (7). In contrast, by updating only on event frames, the model is prevented from forgetting behaviour due to lack of activity.

Illustrative results are presented in the *Camouflage* example. In this case, after a normal model (rightward motion) is acquired, there is a relatively long period of time when no activity takes place (300 frames); nevertheless, when activity resumes anomalous behaviour still is detected relative to the model acquired prior to the no activity period. The benefit is quantified in the associated PR curve in Fig. 2, which compares the proposed method with the same approach neglecting event-based processing. It is seen that event-based processing yields higher precision at comparable recall for any detection threshold,  $\Delta$ , as the model is better maintained. Without event-based processing the activity following the period of no activity consistently is misclassified, as shown in Fig. 5, whereas, with event-based processing it consistently is classified correctly. Nevertheless, the approach still allows for the model to encompass newly recurring behaviours (e.g. moving shadows/lights in *Train*), according to the update rule, (7).

## 4 Discussion

This paper has presented a novel approach to detection of anomalous behaviour in temporal image sequences. The approach centres around three key ideas. First, imagery is represented in terms of distributions of spatiotemporal oriented energy to model normal behaviour as well as record new observations. This representation allows the approach to capture a wide range of naturally occurring behaviours while making fine grained distinctions between model and new observation with robustness to variations in illumination and purely spatial appearance. Second, model and observations are compared via histogram subset inclusion matching. Subset inclusion matching allows for partial matches between model and observation so that not every possible modeled activity must occur at any given time instance to avoid being considered anomalous. Third, event driven processing is employed to allow for focusing of computational effort on portions of the image stream where anomalies might occur. A limitation of the current approach is that it does not explicitly account for non-local phenomena (e.g., interactions between separate local measurements in space and time). Future work will extend the approach to deal with such matters, e.g., by overlaying a MRF on the approach's local observations to abstract interactions.

The entire approach has been instantiated in implementations that show real-time performance. In empirical evaluation, the implementations yield strong performance in being able to model a wide range of potentially complicated patterns of normal activity and detect fine deviations from that model, even while being robust to changes that are insignificant (e.g., illumination and spatial appearance variations). Various compared alternative approaches were not able to yield comparatively strong results.

## References

1. <http://www.cse.yorku.ca/vision/research/anomalous-behaviour>
2. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. *PAMI* 30, 555–560 (2008)
3. Affifi, A., Azen, S.: *Statistical Analysis*. Academic (1979)
4. Andrade, E., Blunsden, S., Fisher, R.: Modelling crowd scenes for event detection. In: *ICPR*, pp. 175–178 (2006)
5. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: *CVPR* (2008)
6. Bebezeth, Y., Jodoin, P., Saligrama, V., Rosenberger, C.: Abnormal events detection based on spatio-temporal co-occurrences. In: *CVPR*, pp. 2458–2465 (2009)
7. Black, M.: Explaining optical flow events with parameterized spatio-temporal models. In: *CVPR*, pp. 326–332 (1999)
8. Boiman, O., Irani, M.: Detecting irregularities in images and in video. *IJCV* 74, 17–31 (2007)
9. Buxton, H.: Learning and understanding dynamic scene activity: A review. *IVC* 23 (2003)
10. Chomat, O., Crowley, J.: Probabilistic recognition of activity using local appearance. In: *CVPR*, pp. 104–109 (September 1999)
11. Derpanis, K., Sizintsev, M., Cannons, K., Wildes, R.: Efficient action spotting based on a spacetime oriented structure representation. In: *CVPR* (2010)
12. Derpanis, K., Wildes, R.: Early spatiotemporal grouping with a distributed oriented energy representation. In: *CVPR* (June 2009)

13. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behaviour recognition via sparse spatio-temporal features. In: PETS, pp. 65–72 (2005)
14. Elgammal, A., Durauswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density for visual surveillance. *Proc. IEEE* 90, 1151–1163 (2002)
15. Freeman, W., Adelson, E.: Design and use of steerable filters. *PAMI* 13, 891–906 (1991)
16. Granlund, G., Knuttson, H.: *Signal Processing for Computer Vision*. Kluwer, Dordrecht (1995)
17. Heikkilä, M., Pietkainin, M.: A texture-based method for modeling the background and detecting moving objects. *PAMI* 28, 657–662 (2006)
18. Hu, W., Xian, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: System for learning statistical motion patterns. *PAMI* 28, 1450–1464 (2006)
19. Jahne, B.: *Digital Image Processing*. Springer, Heidelberg (2005)
20. Javed, O., Shafique, K., Shah, M.: A hierarchical approach to robust background subtraction using color and gradient information. In: *Motion Workshop*, pp. 22–27 (2003)
21. Jodoin, P.M., Konrad, J., Saligrama, V.: Modeling background activity for behavior subtraction. In: *ICDSC*, pp. 1–10 (2008)
22. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. *IVC* 14, 609–615 (1996)
23. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: *ICCV* (2007)
24. Kim, J., Grauman, K.: Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In: *CVPR*, pp. 2921–2929 (2009)
25. Kratz, L., Nishino, K.: Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: *CVPR*, pp. 1446–1453 (2009)
26. Li, L., Gong, S., Xiang, T.: Global behaviour inference using probabilistic latent semantic analysis. In: *BMVC* (2008)
27. McIvor, A.: Background subtraction techniques. In: *Proc. Vid. And Img. Comp.*, New Zealand (2000)
28. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using a social force model. In: *CVPR* (2009)
29. Mittal, A., Monnet, A., Paragios, N.: Scene modeling and change detection in dynamic scenes: A subspace approach. *CVIU* 113, 63–79 (2009)
30. Pless, R.: Spatio-temporal background models for outdoor surveillance. In: *EURASIP* (2005)
31. Stauffer, C., Grimson, E.: Learning patterns of activity using real-time tracking. *PAMI* 22, 747–757 (2000)
32. Wang, X., Ma, X., Grimson, E.: Unsupervised activity perception by hierarchical bayesian models. In: *CVPR* (2007)
33. Watson, B., Ahumada, A.: A look at motion in the frequency domain. In: *Motion Workshop*, pp. 1–10 (1983)
34. Wildes, R., Bergen, J.: Qualitative spatiotemporal analysis using an oriented energy representation. In: Vernon, D. (ed.) *ECCV 2000*. LNCS, vol. 1843, pp. 768–784. Springer, Heidelberg (2000)
35. Zhong, H., Shi, J., Visontai, M.: Detecting unusual activity in video. In: *CVPR* (2004)
36. Zhong, J., Sclaroff, S.: Segmenting foreground objects from a dynamic textured background using a robust Kalman filter. In: *ICCV*, pp. 44–50 (2003)

# Tracklet Descriptors for Action Modeling and Video Analysis

Michalis Raptis and Stefano Soatto

University of California, Los Angeles  
{mraptis, soatto}@cs.ucla.edu

**Abstract.** We present spatio-temporal feature descriptors that can be inferred from video and used as building blocks in action recognition systems. They capture the evolution of “elementary action elements” under a set of assumptions on the image-formation model and are designed to be insensitive to nuisance variability (absolute position, contrast), while retaining discriminative statistics due to the fine-scale motion and the local shape in compact regions of the image. Despite their simplicity, these descriptors, used in conjunction with basic classifiers, attain state of the art performance in the recognition of actions in benchmark datasets.

## 1 Introduction

The analysis of “activities” (or “events” or “actions”) in video is important and yet elusive as there is no obvious taxonomy and their measurable correlates are subject to significant variability. While many activities can be classified based on still images [33], the temporal evolution is important to tease apart more subtle differences [12]; it is obvious that a viable approach has to successfully combine both spatial and temporal statistics. We use the words “activities” or “actions” in quotes, because we do not have a precise (operational) definition for them. However, we postulate that such complex phenomena can be understood as the composition of relatively simple *spatio-temporal statistics*, which we will attempt to characterize in Sect. 2.

In this paper we define elementary spatio-temporal statistics under a set of modeling assumptions about the image formation process (Sect. 2), propose a model to infer them (Sect. 2.2), and evaluate the resulting descriptors on classification tasks using benchmark datasets (Sect. 4).

We *focus on low-level representation*, to devise statistics of the spatio-temporal signal that are insensitive to nuisance factors and yet sufficiently discriminative, that can be used as *building blocks* for more sophisticated models that exploit top-down structure and priors. Thus we purposefully operate with impoverished models that emphasize the low level, keeping top-down processing, shape and motion priors, and learning machinery to a minimum. Even with this impoverished representation, we show that we can achieve competitive performance in end-to-end classification tasks on benchmark datasets. More importantly, however, we believe that our features can be profitably used by more sophisticated models that do exploit top-down information in the form of global temporal statistics or spatial context.

## 1.1 Related Work

We propose spatio-temporal feature descriptors that capture the local structure of the image around trajectories tracked over time. We actively *restrict* our attention to a subset of the spatial image domain and encode its “local photometry”. Our approach differs from “holistic” ones [3, 8, 43, 20, 42] that use the entire video volume to extract global statistics, and compare them with standard norms, block correlation [43], or dynamic time warping [20]. Unlike these approaches, we explicitly model “simple” nuisance variability (position, contrast etc.), detect a corresponding frame with a co-variant detector, and “undo” it in the descriptor, which is therefore by construction invariant to such nuisances. The residual “complex” nuisances (local deformation, deviation from Lambertian reflection, complex illumination changes) are instead averaged out in the descriptor. Such averaging is performed relative to the structure of the nuisances, learned during the training phase, and plays a similar role to spatial binning (a form of “unstructured” averaging) in [23]. In this sense, our approach relates to part-based representations for action recognition, including [34, 7, 21, 29, 40].

Different local descriptors have been proposed to capture shape [34, 7] or joint motion and shape [18, 17, 4] by aggregating features within video cubes centered at spatio-temporal interest points into a static descriptor. In contrast, we retain in our *tracklet* descriptor the entire feature time series from birth to death of each tracked region. Other recent works [38, 27, 25, 13] also use a collection of trajectories to increase the discriminative power of local spatio-temporal volumes, but utilize different representations: [38] uses the stationary statistics of the Markov chain of instantaneous velocities to describe the evolution of the trajectories, which suffers from small-sample effects, while we explicitly maintain the entire time series and employ dynamic time warping to compare our variable-length descriptors. Messing *et al.* [27] use velocities as observations in a sequential graphical model.

We illustrate the general architecture of our descriptors using off-the-shelf detectors and local motion estimators and perform averaging or aggregation using the computational architecture of [23]. While more sophisticated instantiations are possible, already these simple choices attain state-of-the-art performance in the Activities of Daily Living (ADL)[27], the KTH [34] and the Hollywood Human Action (HOHA) [17] datasets. The implementation of the proposed descriptor is available at: <http://vision.ucla.edu/~raptis/tracklets>.

## 2 Spatio-temporal Tracklet Descriptors

We now describe the modeling assumptions under which we operate, and the procedure to infer the resulting representation (Sect. 2.2). While one would want to assemble these elementary actions (dictionary elements) into a model that captures the joint spatio-temporal statistics at a more global spatial scale (“context”), in Sect. 4 we show that even a naive use of the dictionary labels as a “spatial bag” yields competitive performance in end-to-end tasks.

## 2.1 Model and Assumptions

We assume that each “object” is defined *at rest* as a compact region of space, only part of which may be visible due to occlusions, and projected onto a subset  $D$  of the image plane, yielding a function  $\rho : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^+$ ;  $x \mapsto \rho(x)$  where  $D \subset \Omega$  is the *base image region*. There is no requirement that an entire object be captured by one base region. Instead, we can expect objects to be over-segmented in multiple base regions, with their spatio-temporal relations characterizing the object.<sup>1</sup> Base regions move under the action of a finite-dimensional group  $g(t) \in \mathbb{G}$ , which we assume without loss of generality to be the group of rigid motions  $\mathbb{G} = SE(2)$ , with the residual motion, that depends on the shape of the scene and viewpoint, captured by a general diffeomorphism  $w : \Omega \rightarrow \Omega; x \mapsto w(x)$ . Finally, a contrast transformation is applied to the range of the image in the base region, and all other photometric factors (specularities, translucency, inter-reflections etc.) are lumped together as an additive component  $n(x, t)$ . These assumptions are summarized in the model:

$$\begin{cases} \rho(x), x \in D \subset \mathbb{R}^2 & \text{base region} \\ \rho \circ g(t) \doteq \rho(g(t)x), g(t) \in SE(2) & \text{global motion} \\ \rho \circ w(x, t) \circ g \doteq \rho(w(g(t)x, t)) \quad w : \mathbb{R}^2 \rightarrow \mathbb{R}^2 & \text{local deformation} \\ h(t) \circ \rho \circ w \circ g \doteq h(\rho(w(g(t)x, t)), t), \quad h : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ & \text{contrast} \\ I(x, t) = (h \circ \rho \circ w \circ g)(x, t) + n(x, t) & \text{complex illumination, noise, etc.} \end{cases} \quad (1)$$

The above equation is valid only for those  $x \in \mathbb{R}^2$  that intersect the domain of the image  $\Omega$ . Elsewhere, the image is due to phenomena other than the base region, which we call *clutter*,  $\beta(x, t)$ . So, the actual measured image is given by

$$I(x, t) = \begin{cases} h \circ \rho \circ w \circ g(x, t) + n(x, t), & \forall x \in g^{-1}(t)w^{-1}(D, t) \cap \Omega \\ \beta(x, t) & \text{elsewhere.} \end{cases} \quad (2)$$

The *components* (hidden factors) of the extended temporal observation of an object are the (multiple) base image regions  $\rho|_D^i$ , their (variable) length  $\hat{T}_i = T_i - \tau_i$ , global trajectory  $\{g_i(t)\}_{t=\tau_i}^{T_i}$ , their local deformation<sup>2</sup>  $\{w_i(x, t); x \in g_i(t)D\}_{t=\tau_i}^{T_i}$ , the contrast transformation  $\{h_i(t)\}_{t=\tau_i}^{T_i}$ , while everything else is lumped in  $n_i(x, t)$ . In the rest of this section we will omit the index  $i$  and focus on *inference and representation*: How can we “extract” the hidden components from a time series  $\{I(x, t), x \in \Omega\}_{t=\tau}^T$ ? What components of the data-formation process matter for classification? In order to make the inference tractable, we make the following *modeling assumptions*: The effect of complex nuisances  $n(x, t)$  is small relative to other factors, so we **(a)** seek explanations of the data that minimize their effects (e.g. a suitable norm of  $n(x, t)$ ). The contrast

<sup>1</sup> Although it is precisely these contextual spatial relations that we ignore in Sect. 4, to test the representational power of the descriptor alone.

<sup>2</sup> Here  $g_i D = \{g_i x \mid x \in D\}$ .



transformation  $h$  “contains no information” (i.e., we wish the outcome of the task to be independent of contrast), so we **(b)** seek to eliminate it from the representation. The global motion  $g(t)$  *may or may not* contain information, depending on the task, so we **(c)** seek to infer it from the data for later use, or to **(d)** provide a local reference where to compute the deformation field  $w(x, t)$ . The base region  $\rho$  and the local deformation  $w$  contain all the photometric, geometric and dynamic information, respectively, embedded in the data. Therefore, the *inference problem* can be stated as:

$$\{\hat{\rho}, \hat{w}, \hat{g}\}_{t=\tau}^T = \arg \min_{\rho, w, D, h, g} \int_{\tau}^T \|n(x, t)\|_D dt \tag{3}$$

subject to (2), where  $\|n(x, t)\|_D = \int_D |n(x, t)|^2 dx$ , with the addition of an area regularizer to avoid the trivial solution  $D = \emptyset$ . This formalizes (a). To eliminate  $h$ , (b) we simply encode the estimate of the base image region  $\hat{\rho}_I(x) \doteq I \circ \hat{w}^{-1} \circ \hat{g}^{-1}$  using a complete contrast-invariant, such as the geometry of the level lines (or its dual, the gradient orientation), or a local contrast normalization, e.g.

$$\phi(\hat{\rho}(x)) \doteq \frac{\nabla \hat{\rho}_I(x)}{\|\nabla \hat{\rho}_I(x)\|_{\epsilon}} \quad \text{or} \quad \phi(\hat{\rho}(x)) \doteq \frac{I - \int_D I dx}{\|\text{std}(I|_D)\|_{\epsilon}} \tag{4}$$

where<sup>3</sup>  $\|I\|_{\epsilon} = \min\{\|I\|, \epsilon\}$ . We are then left with estimating (c) the global motion  $g$ , and (d) the local deformation  $w$ . Rewriting eq. (3) we have a sequence of equivalent optimizations in fewer and fewer unknowns:

$$\begin{aligned} & \arg \min_{h, \rho, w, g} \int_{\tau}^T \int_D |I(x, t) - h \circ \rho \circ w \circ g| dx dt = \text{(thm. 7.4, p. 269 of [31])} \\ & = \arg \min_{\rho, w, g} \int_{\tau}^T \int_D |\phi(I(x, t)) - \phi(\rho \circ w \circ g)| dx dt = \text{(thm. 1, p. 4 of [37])} \\ & = \arg \min_{w, g} \int_{\tau}^T \int_D |\phi(I(x, t)) - \phi(I(x, t + 1) \circ w \circ g)| dx dt \doteq \{\hat{g}(t), \hat{w}(x, t)\} \tag{5} \end{aligned}$$

This problem can be solved using variational optimization techniques [37]; a more efficient, albeit suboptimal, solution can be arrived at by first assuming  $w(x, t) = x$  and estimating  $\hat{g}(t) = \arg \min_g \int_{\tau}^T \int_D |\phi(I(x, t)) - \phi(I(x, t + 1) \circ g(t))| dx dt$  with any tracking algorithm [24, 35, 27]. Then, given  $\{\hat{g}(t)\}_{t=\tau}^T$ , estimate  $\hat{w}(x, t) = \arg \min_w \int_{\tau}^T \int_D |\phi(I(x, t)) - \phi(I(x, t + 1) \circ w \circ \hat{g}(t))| dx dt$  with any optical flow algorithm. Note that  $\hat{w}$  depends on  $\hat{g}$ , and there is no guarantee that substituting  $\hat{w}, \hat{g}$  in (5) minimizes the cost. However, this approach is sufficient for our purposes, otherwise one can revert to an infinite-dimensional optimization of (5).

<sup>3</sup> Since the gradient direction will be weighted by its norm in the averaging operation to compute the descriptor (Sect. 2.2), the value of  $\epsilon$  does not matter in practice. As an alternative, when color images are available, one can use spectral ratios or local normalization to eliminate contrast transformations.

## 2.2 Simplest Instantiation and Inference of the Representation

Following the derivation above, given a video sequence  $\{I(x, t), x \in \Omega\}_{t=1}^T$ , we first select candidate regions via any feature detector [23, 10, 1], and track them over time using a contrast-compensated translational tracker to obtain a number of trajectories  $\{\hat{g}_i(t)\}_{t=\tau_i}^{T_i}$  of varying length  $\hat{T}_i$ , addressing (c). Many trackers also provide a rotational and scale reference; the latter can be used to select the base regions  $D_i \subset \mathbb{R}^2$ . The former can be used to fix local orientation, although we select the vertical image coordinate as reference. In the resulting local frame  $\{D_i, \hat{g}_i(t)\}$  we then estimate the local motion  $\{\hat{w}_i(x, t)\}_{t=\tau_i}^{T_i}$  using any of a number of local optical flow algorithms, the simplest being [24]. This addresses (d) and completes the (co-variant) frame selection process. Therefore, we design an invariant descriptor by representing the image in the selected frame,  $\{D_i, \hat{g}_i(t)\}$  via the contrast invariant  $\{\phi(I \circ \hat{g}_i)\}$ , and concatenate that with the motion field  $\{\hat{w}_i(x, t) \circ \hat{g}_i(t)\}$  in the base region  $D_i$ .

If we had priors on the intra-class variability  $dP(g, w)$ , we would marginalize the resulting descriptor; in their absence, it is common to assume that the object or category of interest is described by an ‘‘uncertainty ball’’ around a reference descriptor, that is therefore ‘‘blurred’’ in some sense, ideally by averaging with respect to the prior, but more often by coarse spatial binning. In the latter case, the descriptor for  $\{\phi(I \circ \hat{g}_i)\}$  corresponds to a histogram of gradient orientations (HoG) [23, 6], and the descriptor for  $\{\hat{w}_i(x, t) \circ \hat{g}_i(t)|_{D_i}\}$  corresponds to a histogram of optical flow vectors (HoF).

Although many have used HoG/HoF descriptors [18, 22, 17, 4], they aggregate them into a static signature, whereas our previous analysis and [36] suggest retaining their temporal evolution. However, rather than averaging by spatial binning (that presumes ergodicity), we prefer to use at least a crude approximation of the prior  $dP(g, w)$  in the form of samples  $\{g(t_j)\}$ ,  $\{w(x, t_j)\}$  inferred during the training phase. The resulting descriptor, which we call AoG (average of gradient orientation) and AoF (average of optical flow), averages over the training samples – inferred in a sliding temporal window  $\{t_j\}_{j=1}^L$  and thought of as samples from an importance distribution:

$$AoG(t|x, g_i, D_i) = \sum_{\tau=t-\lfloor L/2 \rfloor}^{t+\lfloor L/2 \rfloor} \phi(I(x, \tau)) \circ g_i^{-1}(\tau) \quad x \in g_i(\tau)D_i \cap \Omega \quad (6)$$

where  $g_i D_i$  is defined in footnote 2. Although ‘‘oG’’ in AoG stands for the gradient orientation, in analogy to HoG, any other contrast-normalizing statistic  $\phi$  can be used, as in (4). Similarly, we have

$$AoF(t|x, g_i, D_i) = \sum_{\tau=t-\lfloor L/2 \rfloor}^{t+\lfloor L/2 \rfloor} (w_i \circ g_i)(x, \tau) \quad x \in D_i \cap \Omega \quad (7)$$

We call *Tracklet Descriptor* (TD) the concatenation of the entire time series of either HoG/HoF, or AoG-HoF, and compare the two in Sect. 4, where we show

the latter to yield marginally improved performance at a significantly lower computational cost. Optionally, the TD can be augmented with some sample statistic, for instance the trajectory relative to the spatial or spatio-temporal mean.

$$\boxed{\pi_i(t|I) \doteq \{A/HoG_i(t), A/HoF_i(t)\}} \quad (8)$$

As stated in Sect. 1, we postulate *compositionality* of our representation, so it is natural to organize tracklet descriptors into a “dictionary.” However, because we retain the entire time series, the process is more involved as descriptors of different length have to be compared. In Sect. 3 we describe how this can be done using dynamic time warping and clustering by affinity propagation. As an alternative to averaging, one could consider histograms aggregated over time, rather than space, with similar results, as advocated by [19].

### 3 Implementation

Following Sect. 2.2, we reduce the group  $\mathbb{G} = \mathbb{R}^2$  to pure translations, and estimate  $\{\hat{g}_i(t) \in \mathbb{G}\}_{t=\tau_i}^{T_i}$  using [35], as implemented by [2], without affine consistency check, similar to [27]. Features lost during tracking are replaced by newly selected ones. We prune tracks that are less than  $T_i = 5$ -frames long, or that move less than  $\hat{g}_i(T_i) = 3$ -pixels in standard deviation. Unlike [38], we do not impose an upper bound on  $\hat{T}_i$ , and unlike [7, 34, 4, 25] we do not use a fixed time-scale.

#### 3.1 Constructing Tracklet Descriptors

We capture the contrast-invariant statistics  $\phi$  of the base regions  $D_i$  using the gradient orientation spatially binned (HoG) or averaged (AoG) in a sliding temporal window, e.g.,  $L = 5$  with fixed scale and orientation, centered at each spatial location  $\hat{g}_i(t)$  along the trajectory. The size of  $D_i(t)$  could be adapted using the scale component estimated on-line by the tracker. Although we estimate rotation of the base regions  $D_i$  we discard it, and use the vertical component of the image plane as a reference. In yet a simpler instantiation, one can consider the base regions  $D_i$  fixed to, say,  $18 \times 18$  or  $32 \times 32$  pixels. We estimate the local deformation  $\hat{w}_i(x, t)$  using [24] and aggregate it either in a spatial histogram (HoF) or in an average (AoF) within each region  $D_i$ . While HoG/HoF result in a fixed 128-dimensional vector each, AoG/AoF have variable size depending on  $|D_i|$ ; therefore they are quantized into a comparable number of components (225 in the experiments, corresponding to  $15 \times 15$  patches). The two vectors are concatenated<sup>4</sup> and stacked sequentially over time into a matrix.

#### 3.2 Tracklet Dictionary

For each base image region  $D_i$ , a tracklet descriptor represents a multi-dimensional time series,  $\pi_i : [\tau_i, T_i] \rightarrow \mathbb{R}^N$ . To define a distance between two descriptors we

<sup>4</sup> Although one could introduce weights between the spatial and temporal component, and optimize the weight to a particular dataset, we do not do so in Sect. 4.

must discount initial time, speed of execution, and duration of an action. Therefore, we adopt the dynamic time warping (DTW) distance [32]:

$$d(\pi_i, \pi_j) \doteq \inf_{\alpha, \beta \in \mathcal{H}} \frac{1}{M} \sum_{t=1}^M \|\pi_i(\alpha(t)) - \pi_j(\beta(t))\|_1 \quad (9)$$

where  $\alpha, \beta \in \mathcal{H}$  are continuous monotonic transformations [39, 20] of the temporal domain. For HoG, HoF and AoG we use the  $\ell_1$  distance. Optical flow vectors, however, are not sparse, so  $\ell_2$  should be used instead, allowing small discrepancies. Therefore, AoG and AoF cannot be simply concatenated, but instead separate dictionaries, and combinations of separate kernels, have to be learned. The different structures of AoG and HoF also do not lead to a “meaningful” compact descriptor. To make comparison as fair as possible, in Sect. 4 we test AoG vs. HoG in isolation (Table 3). For a track of 100 frames, HoG takes 13 seconds to be computed (in non-optimized C code), whereas AoG takes 0.6 seconds (in Matlab).

Because of the variable length, many commonly used clustering algorithms (e.g., k-means) are inapplicable to clustering time series. Agglomerative clustering [15] and k-medoids have been used to select cluster centers for time series. We compute pairwise distances among tracklet descriptors, and set the distance to infinity for pairs with length ratio not between 0.5 and 2, since DTW does not provide a meaningful warping path for those cases [30]. We use affinity propagation [9] to cluster and select dictionary elements. This method is efficient due to the sparsity of the initial distance matrix and effective to define discriminative exemplars without the need of multiple random initializations that algorithms like k-centers require. In our experiments the size of the dictionaries was not pre-specified but it was automatically selected by affinity propagation.

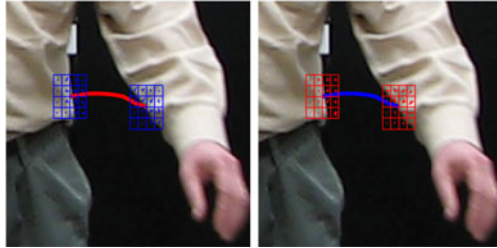
It is not immediate to visualize our cluster centers, since our model is not strictly generative. However, Fig. 1 shows parts of the tracks colored according to their nearest neighbor in a tracklet dictionary. Fig. 2 shows a sample trajectory with samples of the quantized histogram of gradient orientations and optical flow super-imposed on the image. These histograms are concatenated to form a temporal sample of the time series  $\{\pi_i(t)\}$ .

### 3.3 A Basic Classification Scheme

The simplest recognition method we consider is akin to a bag-of-features (BoF) [5], whereby we discard global temporal ordering, capturing only the local temporal variation of a tracklet. This admittedly naive model achieves performance already close to the state of the art. Given a codebook of TDs, we assign each trajectory in a test frame to the closest codebook element (Sect. 3.2); then each video is represented by a histogram of occurrences of dictionary elements. We use a support-vector machine with either a RBF- $\chi^2$  kernel or an intersection kernel. The penalty parameter is selected by 10-fold cross-validation in the training set, whereas the scale parameter of the RBF kernel is selected as the mean  $\chi^2$  distance of the training samples. The RBF- $\chi^2$  SVM achieves an improvement of 1 – 2% over the intersection one.



**Fig. 1.** Tracks extracted from ADL, KTH and HOHA datasets. Color indicates their label based to the tracklet descriptor dictionary.



**Fig. 2.** A track with samples of the histogram of gradient orientation (left, blue) and histogram of optical flow (right, red) along the trajectory. These are concatenated to form a 256-dimensional temporal sample of the time series that represents that elementary action.

## 4 Experimental Evaluation

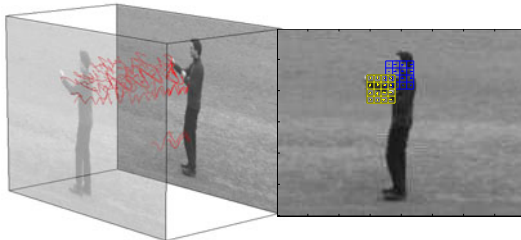
We evaluate the proposed scheme on three publicly available datasets: KTH [34] Activities of Daily Living (ADL) [27] and Hollywood Human Actions (HOHA) [17]. As pointed out in Sect. 3.2, AoG cannot be simply concatenated with either AoF or HoF, but has to be combined using multiple kernels. In our first two experiments we use the compact tracklet descriptor based on the HoG/HoF, so we can use one dictionary and one kernel, and have a fair comparison with existing local descriptors [7]. In the most challenging dataset (HOHA) the individual components HoG and AoG are compared in Table 3, and their combination with HoF is reported in Table 4.

**KTH** is chosen because of its popularity, though its modest spatial ( $160 \times 120$  pixels) and temporal (25 frames per second) resolution make for an impoverished data stream that is not well suited for local representations. There are 6 actions performed by 25 subjects in 4 scenarios (outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoor (s4)), resulting in 598 clips. The simplicity of these actions, combined with an uncluttered static background, make this dataset ideally suited for global representations [20]. Nevertheless, even without exploiting background subtraction or the global evolution

of the silhouette (hard to obtain in most realistic scenarios), our scheme is competitive with the state of the art (Table 1).

More specifically, we track an average of 340 trajectories per video with an average length  $\hat{T}_i = 23$  frames. Low resolution and the compression artifacts are a challenge to tracking, so the average length is relatively small. Our base regions  $D_i$  are fixed at  $18 \times 18$  pixels, similar to the spatial size of Cuboids [7, 28, 29]. Examples of tracks and the corresponding HoG descriptors are shown in Fig. 3. The classification performance of algorithms that use spatio-temporal descriptors computed in volumes around interest points [21, 17, 4, 29, 7, 28] has proven that the choice of the temporal scale is crucial. Laptev *et al.* [17] construct static HoG/HoF around points detected by spatio-temporal Harris-3D [16] at multiple scales, using  $\Delta t = 25, 36$ ; [4] computes a HoG/HoF around points detected by [23] in a volume with  $\Delta t = 60$ . Instead, our descriptors have variable temporal length depending on the image region  $D_i$ . Moreover, the optical flow in the image regions  $D_i$  can be estimated reliably. This is not the case for the spatio-temporal cubes around a specific interest point.

We use leave-one(person)-out cross validation and average the results over the 25 permutations. To construct the codebook we use a relatively small training set, similar to [28], to examine the generalization of our algorithm. We only use the descriptors extracted from the first two parts of the 72 videos of 3 subjects. Those descriptors are excluded from the test and training sets. It should be noted that [21, 4] used the videos of 24 subjects to construct the codebook, whereas [17] used 8 subjects. Using a codebook with 1560 TDs of HoG/HoF, we achieve 94.5% recognition rate using RBF- $\chi^2$  SVM (Table 1) considering the dataset as a single large set (all average in one). Using linear SVM with intersection kernel we achieve 93.82% recognition rate. Considering each scenario separately the recognition rate is : (s1) 98%, (s2) 92.67%, (s3) 91.95% , (s4) 96.67%.



**Fig. 3.** Example of the tracks and an instance of the corresponding appearance descriptor of a boxing action on the KTH dataset

We could push the performance of our algorithm by optimizing the weights between the different components of the features (spatial, motion), but our point is not to propose an action recognition system, but just to evaluate descriptors, so we refrain.

The **ADL** dataset has higher-resolution ( $1280 \times 720$  pixels at 30FPS) with 10 different complex activities targeted to an assisted living scenario (e.g.

**Table 1.** Performance comparison on KTH dataset. Despite not using background subtraction or structural information, our approach is competitive with the state of the art.

	evaluation	Recognition Rate		Structural Information
		all scenarios in one	average of all scenarios	
<b>Our tracklets</b>	Leave-One-Out	<b>94.5%</b>	<b>94.8%</b>	No
Niebles <i>et al.</i> [28]	Leave-One-Out	81.5%	N/A	No
Dollár <i>et al.</i> [7]	Leave-One-Out	81.2%	N/A	No
Schuldt <i>et al.</i> [34]	Split	71.7%	N/A	No
Nowozin <i>et al.</i> [29]	Split	84.7%	N/A	No
Liu <i>et al.</i> [22]	Leave-One-Out	N/A	94.15%	Yes
Lin <i>et al.</i> [20]	Leave-One-Out	93.4%	<b>95.8%</b>	Yes
Messing <i>et al.</i> [27]	Split	74%	N/A	No
Yao <i>et al.</i> [41]	Split	87.8%	N/A	Yes
Laptev <i>et al.</i> [17]	Split	91.8%	N/A	Yes
Jhuang <i>et al.</i> [11]	Split	N/A	91.7%	No
Schindler <i>et al.</i> [33]	Split	92.7%	90.7%	No
Yeffet <i>et al.</i> [42]	Split	90.1%	N/A	Yes
Chen <i>et al.</i> [4]	Leave-One-Out	95.0%	N/A	No

“answering phone (aP),” “eating snack (eS),” “eating banana (eB)”). Five subjects perform each activity thrice for a total of 150 clips of duration varying between 10 and 60 seconds. It has drawbacks similar to KTH, in that all actions are taken against a still background from a fixed vantage point, an incentive to overfitting by using background subtraction and global statistics such as the absolute position of tracks in the image. Despite not using absolute positions, a simple classifier based on TDs HoG/HoF outperforms the state of the art by a sizeable margin. We extract on average 1300 tracks with mean duration  $\bar{T}_i = 110$  frames. The base regions  $D_i$  are fixed at  $36 \times 36$  pixels. We again use leave-one (person)-out evaluation, similar to [27, 26], and report the average over the 5 permutations of the dataset. We randomly sampled 25K tracklets from the training set and constructed a dictionary with 2900 elements. Using this dictionary we achieve 82.67% average recognition rate using RBF- $\chi^2$  SVM (Table 2). Comparison to [27] shows that our tracklet descriptor achieves comparable results without using any structural information (relative position or absolute position). It outperforms [27] even when their classifier uses the position of the extracted trajectories relative to the position of the face of the actor. In order to have a fair comparison with existing methods that report results in the ADL dataset, we incorporate a codebook of the absolute position  $(\bar{g}_i(t), \bar{t})$  of the tracks with size 60 obtained using K-means. Combining linearly the two  $\chi^2$  kernels, we achieved 90% average recognition rate. We should note that, although absolute position is relevant in this dataset, and in particular it helps boost the performance of our algorithm as well as [27] significantly, it does so only because all sequences are taken from the same vantage point, in an environment with fixed layout. In general, we advocate *not* using absolute position, even if it improves the performance in this particular dataset.

The **HOHA** dataset overcomes the limitations of ADL and KTH. The dataset contains 430 movie videos ( $240 \times 450$  at 24FPS) with challenging camera motion, rapid scene changes and cluttered and unconstrained background. Moreover, the

**Table 2.** Performance comparison on ADL. Despite not using structural information or background subtraction, we improve the state of the art by a large margin. Using structural information, which we do not advocate, we can further improve recognition rate to 90%, highlighting the limitations of this particular dataset.

	Recognition Rate
<b>Our Tracklets</b>	<b>82.67%</b>
Spatio-temporal cuboids [7] (implemented by [26] )	43%
Velocity Histories [27]	63%
Latent Velocity Histories [27]	67%
Augmented Velocity Histories with Relative Position [26]	72%
Augmented Velocity Histories with Relative and Absolute Position [27]	89%

human actions that are included are not constrained to single actor behaviors, e.g. “Sit down”, but also interactions between humans, e.g. “Kiss”, and objects, e.g. “Get Out of a Car”. We evaluate our trajectory descriptors following the experimental setting proposed by [17], i.e. the test set has 211 videos with 217 labels and the training set has 219 videos with 231 labels (manually annotated). For each action we train a binary classifier and we evaluate our performance with average precision (AP) of the precision/recall curve.

In order to manage the large variability of the image sequences contained in the dataset, features [35] are detected in multiple scales. We extract on average 500 tracks with mean duration  $\hat{T}_i = 51$  frames. For each image region  $D_i$  a HoG, HoF and AoG descriptor is constructed as described in (Sect. 3). First, a dictionary is created for each individual component of our tracklet descriptors and we evaluate its performance using RBF- $\chi^2$  SVM (Table 3). Our TD of optical flow significantly outperforms the HoF proposed by Laptev *et al.* [17], proving to be more robust to background motion and large viewpoint changes. We also note that the performance of TD HoF is slightly worse than the trajectory transition descriptor (TTD) [38], which is combined with spatio-temporal grid to incorporate some structural information in the descriptor. Our TD of AoG outperforms marginally both our TD HoG and the HoG of [17], at a significantly reduced computational cost. Next, we construct our compact HoG/HoF tracklet descriptor and with a codebook with 2220 elements we achieve 32.1% mean average precision (MAP) (Table 4). In order to fuse the TD AoG feature descriptor with TD HoF feature in our classification framework, we build a kernel as a convex combination of their  $\chi^2$  kernels:  $K_{AoG-HoG} = \lambda K_{AoG} + (1 - \lambda) K_{HoF}$ ,  $\lambda$  was selected using cross-validation in the training set. The performance of the obtained kernel is 34.3% MAP. Our TD descriptors outperforms all the local descriptors that have been evaluated in HOHA dataset in a bag-of-features setting [14, 25, 17] and we are competitive with the holistic approach proposed by [42] and the methods that use multi-channel Gaussian kernels [17, 38] for combining the 48 or more channels provided by spatio-temporal grids.

## 5 Discussion

We have presented local spatio-temporal descriptors intended as low-level statistics to be used in action recognition systems. Our descriptors are deduced from an explicit model with all assumptions explicitly stated. They do not involve top-down modeling and can be efficiently learned from data. They can capture



**Table 3.** Performance comparison on HOHA Dataset of Individual components of Descriptors

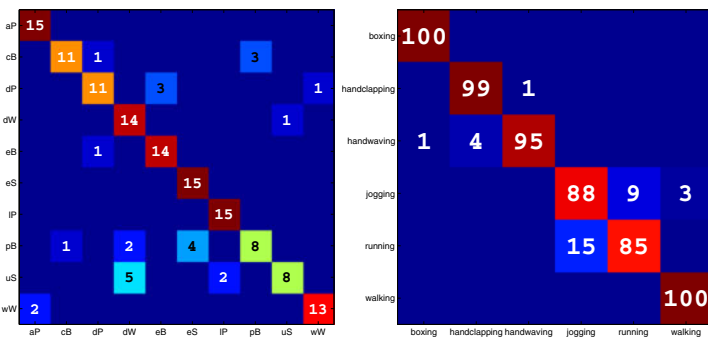
Class	Our Tracklet			Laptev <i>et al.</i> [17]	
	HoG BoF	HoF BoF	AoG BoF	HoG BoF	HoF BoF
Answer phone	24.9%	22.1%	<b>33%</b>	13.4%	<b>24.6%</b>
Get out of car	21.1%	<b>19.3%</b>	<b>22.3%</b>	21.9%	14.9%
Hand shake	<b>20.4%</b>	<b>19.1%</b>	17.4%	18.6%	12.1%
Hug person	22.3%	<b>28.2%</b>	22.0%	<b>29.1%</b>	17.4%
Kiss	48.4%	<b>47.0%</b>	47.5%	<b>52.0%</b>	36.5%
Sit down	21.8%	<b>22.2%</b>	22.5%	<b>29.1%</b>	20.7%
Sit up	<b>16.7%</b>	<b>17.5%</b>	15.3%	6.5%	5.7%
Stand up	40.5%	<b>59.9%</b>	40.2%	<b>45.4%</b>	40.0%
MAP	27.1%	<b>29.4%</b>	<b>27.5%</b>	27.0%	21.5%

**Table 4.** Performance comparison on HOHA Dataset

Class	Our Tracklet		Laptev <i>et al.</i> [17]		Yeffe <i>et al.</i> [42]	Matikainen <i>et al.</i> [25]	Kläser <i>et al.</i> [14]	Sun <i>et al.</i> [38]	
	HoG/HoF BoF	AoG-HoF BoF	Single	Combined				TTD Combined	TTD-SIFT Combined
Answer phone	26.7%	33.0%	26.7%	32.1%	35.1%	35.0%	18.6%		
Get out of car	28.1%	27.0%	22.5%	41.5%	32.0%	7.7%	22.6%		
Hand shake	18.9%	20.1%	23.7%	32.3%	33.8%	5.3%	11.8%		
Hug person	25.0%	34.5%	34.9%	40.6%	28.3%	23.5%	19.8%	N/A	N/A
Kiss	51.5%	53.7%	52.0%	53.3%	57.6%	42.9%	47.0%		
Sit down	23.8%	27.4%	37.8%	38.6%	36.2%	13.6%	32.5%		
Sit up	23.9%	19.0%	15.2%	18.2%	13.1%	11.1%	7.0%		
Stand up	59.1%	60.0%	45.4%	50.5%	58.3%	42.9%	38.0%		
MAP	32.1%	34.3%	32.9%	38.4%	36.8%	22.8%	24.7%	30.3%	44.94%

the discriminative statistics of the local causal structure of the data (temporal ordering), and the local shape and deformation of each base region. However, they do not enforce global shape or motion statistics, nor global temporal ordering. They could be used as a building block of more complex models for the recognition and classification of actions.

Although our goal is not to present a complete action recognition system, in order to test our descriptors we have employed them in simple classification schemes to recognize actions in commonly used benchmark datasets. In all cases, we obtain results comparable to or exceeding the state of the art, despite not making use of top-down structure.

**Fig. 4.** Confusion matrices for ADL dataset (Left) and for KTH dataset (Right)

## References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
2. Birchfield, S.: Klt: An implementation of the kanade-lucas-tomasi feature tracker (1996)
3. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. *IEEE Trans. on Pattern Anal. and Machine Intell.* (2001)
4. Chen, M., Mummert, L., Pillai, P., Hauptmann, A., Sukthankar, R.: Exploiting multi-level parallelism for low-latency activity recognition in streaming video. In: Proc. of the First Annual ACM SIGMM Conf. on Multimedia systems. ACM, New York (2010)
5. Csurka, G., Dance, C.R., Dan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proc. of the Eur. Conf. on Computer Vision, ECCV (2004)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2005)
7. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (October 2005)
8. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. Intl. Conf. on Computer Vision (2003)
9. Frey, B., Dueck, D.: Clustering by passing messages between data points. *Science* 315(5814), 972 (2007)
10. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, Manchester, UK, vol. 15, p. 50 (1988)
11. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: Proc. Intl. Conf. on Computer Vision (2007)
12. Johansson, G.: Visual perception of biological motion and a model for its analysis. *Perceiving events and objects* (1973)
13. Kaâniche, M., Brémond, F.: Gesture recognition by learning local motion signatures. In: Proc. Conf. Computer Vision and Pattern Recognition (2010)
14. Kläser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3dgradients. In: British Machine Vision Conference, September 2008, pp. 995–1004 (2008)
15. Kumar, M., Patel, N., Woo, J.: Clustering seasonality patterns in the presence of errors. In: Proceedings of the Eighth ACM SIGKDD (2002)
16. Laptev, I.: On space-time interest points. *Intl. J. of Comp. Vis.* 64(2), 107–123 (2005)
17. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Proc. Conf. Computer Vision and Pattern Recognition (2008)
18. Laptev, I., Pérez, P.: Retrieving actions in movies. In: Proc. Intl. Conf. on Computer Vision (2007)
19. Lee, T., Soatto, S.: An end-to-end visual recognition system. Technical Report UCLA-CSD-100008 (February 10, 2010) (revised March 18, 2010)
20. Lin, Z., Jiang, Z., Davis, L.: Recognizing actions by shape-motion prototype trees. In: Proc. Intl. Conf. on Computer Vision (2009)
21. Liu, J., Luo, J., Shah, M.: Recognizing Realistic Actions from Videos “in the Wild”. In: Proc. IEEE Computer Vision and Pattern Recognition (2009)

22. Liu, J., Shah, M.: Learning human actions via information maximization. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008)
23. Lowe, D.: Object recognition from local scale-invariant features. In: Intl. Conf. on Computer Vision (1999)
24. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. 7th Int. Joint Conf. on Art. Intell. (1981)
25. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: ICCV workshop on Videooriented Objected and Event Classification (2009)
26. Messing, R., Pal, C.: Behavior recognition in video with extended models of feature velocity dynamics. In: AAAI Spring Symposium Technical Report (2009)
27. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: Intl. Conf. on Computer Vision (2009)
28. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. Intl. J. of Comp. Vis. 79(3) (2008)
29. Nowozin, S., Bakir, G., Tsuda, K.: Discriminative subsequence mining for action classification. In: Proc. Intl. Conf. on Computer Vision (2007)
30. Rabiner, L., Juang, B.: Fundamentals of speech recognition. Prentice Hall, Englewood Cliffs (1993)
31. Robert, C.P.: The Bayesian Choice. Springer, New York (2001)
32. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 26(1), 43–49 (1978)
33. Schindler, K., Van Gool, L.: Action snippets: How many frames does human action recognition require? In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008)
34. Schuldts, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: Proc. Intl. Conf. on Pattern Recognition (2004)
35. Shi, J., Tomasi, C.: Good features to track. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1994)
36. Soatto, S.: Towards a mathematical theory of visual information (2010)
37. Soatto, S., Yezzi, A.: Deformation: deforming motion, shape average and the joint segmentation and registration of images. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 32–47. Springer, Heidelberg (2002)
38. Sun, J., Wu, X., Yan, S., Cheong, L., Chua, T., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2009)
39. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.: The function space of an activity. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2006)
40. Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: British Machine Vision Conference (2009)
41. Yao, B., Zhu, S.: Learning Deformable Action Templates from Cluttered Videos. In: Intl. Conf. on Computer Vision (2009)
42. Yeffet, L., Wolf, L.: Local trinary patterns for human action recognition. In: Proc. Intl. Conf. on Computer Vision (2009)
43. Zelnik-Manor, L., Irani, M.: Statistical analysis of dynamic actions. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(9), 1530–1535 (2006)

# Word Spotting in the Wild

Kai Wang and Serge Belongie

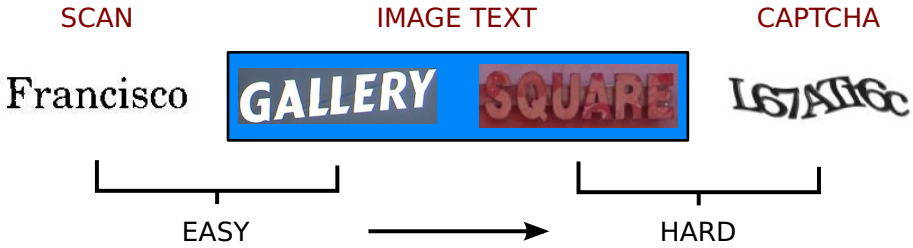
Department of Computer Science and Engineering  
University of California, San Diego  
{kaw006,sjb}@cs.ucsd.edu

**Abstract.** We present a method for spotting words *in the wild*, i.e., in real images taken in unconstrained environments. Text found in the wild has a surprising range of difficulty. At one end of the spectrum, Optical Character Recognition (OCR) applied to scanned pages of well formatted printed text is one of the most successful applications of computer vision to date. At the other extreme lie visual CAPTCHAs – text that is constructed explicitly to fool computer vision algorithms. Both tasks involve recognizing text, yet one is nearly solved while the other remains extremely challenging. In this work, we argue that the appearance of words in the wild spans this range of difficulties and propose a new word recognition approach based on state-of-the-art methods from generic object recognition, in which we consider object categories to be the words themselves. We compare performance of leading OCR engines – one open source and one proprietary – with our new approach on the ICDAR Robust Reading data set and a new word spotting data set we introduce in this paper: the Street View Text data set. We show improvements of up to 16% on the data sets, demonstrating the feasibility of a new approach to a seemingly old problem.

## 1 Introduction

Finding words in images is an fundamental computer vision problem, and is especially challenging when dealing with images acquired in the wild. The field of Optical Character Recognition (OCR) has a long history and has emerged as one of the most successful practical applications of computer vision. However, text found in the wild can take on a great variety of appearances, and in many cases can prove difficult for conventional OCR techniques. Figure 1 shows examples of text on a spectrum of difficulty levels. When we consider the extreme cases, the performance of OCR engines is known to be excellent when given scanned text and very poor on text that is highly obscured. Indeed, the fact that OCR has difficulty reading such text is the basis for systems that prevent automated software bots from abusing internet resources, which are known as CAPTCHAs [1]. Depending on the particular instance, text found in the wild can appear similar to a scanned page, similar to a CAPTCHA, or somewhere in-between.

Our use of the phrase *in the wild* is analogous to Labeled Faces in the Wild (LFW) [2]: a data set constructed to study face recognition in unconstrained

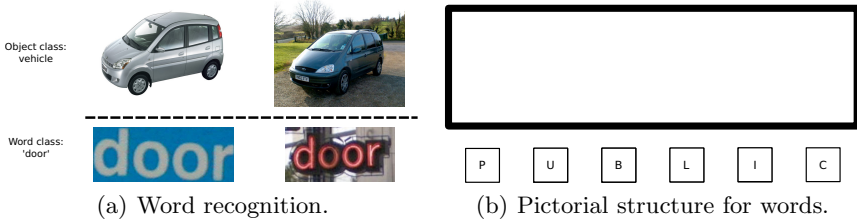


**Fig. 1.** This figure shows examples of images of words ordered by difficulty. In the extreme cases, the behavior of OCR engines is well understood: it is highly accurate when reading scanned text (far left) and is inaccurate when reading a CAPTCHA [1] (far right). In between these two extremes sits text found in the wild. Due to its unconstrained nature, in some cases the image text is similar to scanned text and can be read, while in others it cannot.

settings. Similar to text reading, face recognition under controlled settings is a well understood problem with numerous effective algorithms. However, as LFW shows, the variation in lighting, pose, imaging device, etc., introduce challenges for recognition systems. Much as that dataset acted as a catalyst for renewing progress in face recognition, an important goal of this work is to spur interest in the problem of spotting words in the wild.

The word spotting problem contrasts with general text reading in that the goal is to identify specific words. Ideally, there would be no distinction between the standard text reading and word spotting; spotting words would simply amount to filtering the output from OCR engines to catch the words of interest. However, due to the challenges presented by text found in the wild, we approach the word spotting problem directly, where we are presented with an image and a lexicon of words to spot. We evaluate the performance of conventional OCR engines and also present a new method rooted in ideas from object recognition. In our new approach, we treat each word in a lexicon as an object category and perform word category recognition. Figure 2(a) shows an analogy to generic object recognition: just as instances of the object category *vehicle* can look vastly different from image to image, the word ‘door’ can also take on a variety of appearances depending on the font, lighting, and pose in a scene. In this formulation, we can leverage techniques that have been designed to be robust for recognizing generic categories and apply them to word recognition.

Our contributions are the following. (1) We introduce the Street View Text data set: an outdoor image text data set annotated with a list of local business names per image. (2) We benchmark conventional OCR engines on our new data set and the existing ICDAR Robust Reading image text database [3]. (3) We present a new word spotting approach that imports techniques from generic object recognition and significantly outperforms conventional OCR based methods.



**Fig. 2.** The left figure (a) shows our analogy to the generic object classification problem. In both cases, individual instances of the same class can take on vastly different appearances. The right figure (b) is an illustration of modeling the word ‘PUBLIC’ using a pictorial structure.

## 2 Motivating Applications

Accurate word spotting plays an important role in systems for image retrieval and navigation. Research in Content Based Image Retrieval (CBIR) [4] has explored different forms of querying large image collections, including queries by keyword and image example. Integrating a word spotting component enables queries by word occurrence, returning images in which the specified words appear. The work of [5] describes a system that allows for retrieval of historical documents based on handwritten word spotting.

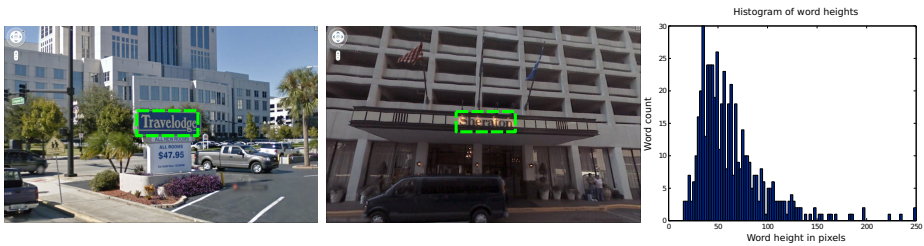
Word spotting is an essential component of a vision based navigation system. In our case, this arises in the form of developing assistive technologies for the blind. Two broad goals of the project are to develop a computer vision system that can benefit the blind and visually impaired communities, and to study the challenges of performing vision-based navigation in real world environments. For navigation, it is important to be able to spot specific keywords in order to guide a blind user. Detecting keywords on signage can be used, for example, to direct a user to the correct aisle in a supermarket while detecting words from a shopping list can be used to locate specific products.

## 3 Dataset

We introduce the Street View Text<sup>1</sup> (SVT) data set harvested from Google Street View<sup>2</sup>. Image text in this data exhibits high variability and often has low resolution. Figure 3 shows examples from the SVT set and a histogram of word heights. In dealing with outdoor street level imagery, we note two characteristics. (1) Image text often comes from business signage and (2) business names are easily available through geographic business searches. These factors make the SVT set uniquely suited for word spotting in the wild: given a street view image, the goal is to identify words from nearby businesses.

<sup>1</sup> <http://vision.ucsd.edu/project/grocr>

<sup>2</sup> <http://maps.google.com>



**Fig. 3.** Examples from our Street View Text (SVT) data set and a histogram of word heights. The words appearing in this data set have high variability in appearance, suffer effects of cast shadows, and often have low resolution. The median height is 55 pixels.

*Data Collection.* We used Amazon’s Mechanical Turk<sup>3</sup> to harvest and label the images from Google Street View. To build the data set, we created several Human Intelligence Tasks (HITs) to be completed on Mechanical Turk. We refer to those that work on these HITs as *workers*.

*Harvest images.* Workers are assigned a unique city and are requested to acquire 20 images that contain text from Google Street view. They were instructed to: (1) perform a *Search Nearby*:\* on their city, (2) examine the businesses in the search results, and (3) look at the associated street view for images containing text from the business name. If words are found, they compose the scene to minimize skew, save a screen shot, and record the business name and address.

*Image annotation.* Workers are presented with an image and a list of candidate words to label with bounding boxes. This contrasts with the ICDAR Robust Reading data set in that we only label words associated with businesses. We used Alex Sorokin’s Annotation Toolkit<sup>4</sup> to support bounding box image annotation. All images were labeled by three workers, and bounding boxes were accepted when at least two workers agreed with sufficient overlap.

For each image, we obtained a list of local business names using the *Search Nearby*:\* in Google Maps at the image’s address. We stored the top 20 business results for each image, typically resulting in 50 unique words. To summarize, the SVT data set consists of images collected from Google Street View, where each image is annotated with bounding boxes around words from businesses around where the image was taken. The data set contains 350 total images (from 20 different cities) and 725 total labeled words. We split the data into a training set of 100 images and test set of 250 images, resulting in 211 and 514 words in the train and test sets. In correspondence with ICDAR, we divide our benchmark into SVT-SPOT (word locating), SVT-WORD (word recognition), and SVT-CHAR (character recognition). In this work, we address SVT-WORD. In total, the cost of acquiring the data from Mechanical Turk was under \$500 USD.

<sup>3</sup> <http://mturk.com>

<sup>4</sup> <http://vision.cs.uiuc.edu/annotation/>

## 4 Related Work

### 4.1 Scanned Document OCR

The topic of OCR has been well studied [6] [7] and existing commercial products are in widespread use. One example is Google Book Search<sup>5</sup>, which has scanned more than 10 million volumes<sup>6</sup>, making them accessible for full text searches. Another example is the Kurzweil National Federation of the Blind (KNFB) reader<sup>7</sup>. The KNFB reader is an OCR engine that runs on a mobile phone and allows a person who is visually impaired to read printed text from an image taken by the camera. The key to high performance for the KNFB reader is having a high quality camera built into the mobile phone and a feedback loop to assist the user in taking pictures in an ideal setting, thereby minimizing the effects of motion blur, lighting, and skew.

A critical step for OCR accuracy is image binarization for character segmentation. The survey of [8] identifies incorrect segmentation as one of the major contributors to errors in using conventional OCR on scanned documents. Previous work on classifying hand written digits from the MNIST data set has shown that when the correct segmentation is provided, it is possible to achieve recognition rates nearing that of humans<sup>8</sup>. The task of separating out individual characters was also identified in [9] as one of the distinguishing features of CAPTCHAs being difficult for OCR while remaining manageable for humans. Character segmentation is a significant challenge that conventional OCR engines face when dealing with words in the wild.

### 4.2 Image Text OCR

OCR in non-scanned images is a relatively new area and has seen increasing attention [10] [11] [12]. Existing work on image text typically breaks the process into two subtasks: text detection and word recognition. Advances have been made in detecting image text using an AdaBoost-based approach [13]. In that work, detected text regions are sent to a conventional OCR engine to be decoded. Others have explored the problem of improving recognition rates by combining outputs of several different OCR engines to get a more robust reading [14].

The works that are most similar to ours are that of [15] and [5]. In [15], the authors investigated methods of breaking visual CAPTCHAs. In their CAPTCHA experiments, the problem was also one of word spotting: categorize the image of a word as one of a list of possible keywords. Our new approach highlights the similarities between words in the wild and with visual CAPTCHAs. In [5], the authors performed word spotting in scanned handwritten historical documents. To perform word spotting, they clustered words together by appearance,

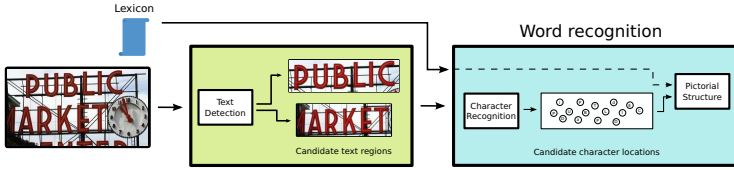
<sup>5</sup> <http://books.google.com/>

<sup>6</sup> <http://googleblog.blogspot.com/2009/10/tale-of-10000000-books.html>

<sup>7</sup> <http://www.knfbreader.com/>

<sup>8</sup> <http://yann.lecun.com/exdb/mnist/index.html>





**Fig. 4. Word spotting overview.** This is an illustration of a word spotting system with two steps: text detection [13] and word recognition. In this work, we focus on the latter problem where the input is an image region and a lexicon of words. In our Street View Text data set, the lexicon was created out of local business searches around where the image was acquired. We run character detectors to discover possible character locations and then score words in our lexicon by modeling them as pictorial structures.

manually provided labels to clusters, and propagated the labels to the cluster members, allowing them to create a word index to browse a large corpus.

In our methods, we draw on work done using part-based methods for object recognition; in particular, the modeling of objects using pictorial structures [16] [17]. We also build on the work of [18], who studied the use of various features and classification methods to classify individually cropped characters.

## 5 Word Recognition

In our approach, we first perform character detection for every letter in an alphabet and evaluate the configuration scores for the words in our lexicon to find the most suitable one. Our method is designed to be used in conjunction with a text detector. In our description, we use the term ‘input image’ to mean the cropped out image region around a word provided by a text detector. Figure 4 shows a diagram of this pipeline.

### 5.1 Character Recognition

Character recognition in images was recently studied in [18]. In their work, they benchmarked different features and classification algorithms for recognizing cropped characters. In our experiments, we test our character detector using the same data and methodology, and list accuracies next to those from their work. For our character detector, we use Histograms of Oriented Gradient (HOG) [19] features with a nearest neighbor classifier.

*Character classification:* To compare two images of cropped characters, we first resize them to take on the same height and aspect ratio, then densely calculate their HOG features. Each character is now represented as an array of dimension  $m \times n \times d$  where  $m$  and  $n$  are the number of rows and columns after spatial binning, and  $d$  is the number of dimensions in each histogram. We measure the similarity between characters by performing Normalized Cross Correlation

(NCC) between each dimension and averaging the scores. Since the characters were resized to be the same dimension, the result is a single number. This is the value we use for nearest neighbor classification.

*Character detection:* To perform character detection over an input image we take all the training examples for a particular character class, resize them to the height of the input image (while maintaining aspect ratio), and compare the character's HOG features to those of the input. Between each training example and the input, we again calculate the NCC between each HOG dimension and combine them again by averaging. The result will be a list of scores measuring the similarity of a template to each location in the input image. This is done for all the training examples of a class, and the results are combined together per class by taking the max at each location. We perform non-maximum suppression to discover peaks and consider those as candidate character locations.

This is done for every character class to create a list of character locations with discrete spatial positions. Next, we use this list of detections to evaluate the configuration of strings in our lexicon to the input image.

## 5.2 Word Configuration

After performing character detection, we consider each word in our lexicon and measure its character configuration within the input image. We represent a word using a pictorial structure [16] [17]. A pictorial structure is a mass-spring model that takes into account costs of matching individual parts to image locations and their relative placement. A word is naturally broken down into character 'parts' and takes on a simple chain structure. Figure 2(b) shows an example of a string as a pictorial structure.

We formulate the problem of optimal character placement in an image of text in the following way. Let  $G = (V, E)$  be an undirected graph representing a string  $S$ . The vertices  $V = \{v_1, \dots, v_n\}$  correspond to characters in  $S$  where  $n$  is the length of  $S$ . Edges  $(v_i, v_j) \in E$  connect letters that are adjacent in  $S$ . This creates a conceptual spring between pairs of letters. We use the terms parent and child to refer to the left and right nodes in a pair of adjacent characters. Let  $L = (l_1, \dots, l_n)$  represent a particular configuration of characters in an image where  $l_i$  is the spatial  $[x, y]^T$  coordinate placement of character  $v_i$ .

We measure cost  $m_i(l_i)$  as one minus the similarity score of a character detection calculated in the previous step. To calculate the deformation cost  $d_{i,j}(l_i, l_j)$ , we use our domain knowledge of character layout. We expect a child character to appear one character width away from its parent. Let the expressions  $w(l_i)$  and  $h(l_i)$  represent the width and height of a character detection at location  $l_i$ . Let  $l_i^* = l_i + [w(l_i), 0]^T$  represent the expected position of a child of  $l_i$ . We specify a covariance matrix that normalizes the deformation cost to the dimensions of the parent character:  $\Sigma = \begin{bmatrix} w(l_i) & 0 \\ 0 & h(l_i) \end{bmatrix}$ . Our deformation cost is calculated

as:  $d_{i,j}(l_i, l_j) = \sqrt{(l_i^* - l_j)^T \Sigma^{-1} (l_i^* - l_j)}$ . The objective function for our optimal character configuration for a string  $S$  is computed as:

$$L^* = \arg \min_L \left( \theta \sum_{i=1}^n m_i(l_i) + (1 - \theta) \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right) \quad (1)$$

The parameter  $\theta$  controls the balance between part match cost and deformation cost. The result is a configuration  $L^*$  that represents the optimal character placement for reading  $S$  in an image. Solving for  $L^*$  can be done efficiently using dynamic programming as described in [17]. We refer to this configuration cost as  $D_c(L)$ .

The score generated by  $L^*$  can take into account a local measure of coherence between a string and an image, but is uninformed of higher order and global configuration costs. To supplement the score configuration score, we also incorporate other domain knowledge-influenced measures into our final match score.

- **Horizontal span:** Given our input is an image of a cropped word from a character detector, we assume that a suitable string is one whose characters span most of the input image. We calculate this as the horizontal range of the character configurations divided by the width of the input image and call it  $D_s(L)$ .
- **Character distribution:** Character spacing within a single string should be consistent, and we factor this into the final score by measuring the standard deviation of the spacing between every pair of adjacent characters in the string, which we refer to as  $D_d(L)$ .

The final cost  $D$  is a weighted sum of these terms:  $D(L) = \alpha_1 D_c(L) + \alpha_2 D_s(L) + \alpha_3 D_d(L)$  where  $\alpha_1 + \alpha_2 + \alpha_3 = 1$ . Through validation on our training data, we determined reasonable parameters to be  $\theta = .9$ ,  $\alpha_1 = .5$ ,  $\alpha_2 = .4$ , and  $\alpha_3 = .1$ . These parameters were used in both the ICDAR and SVT benchmarks.

## 6 Experiments

We evaluate the performance of our character recognizer in isolation and our word recognition system as a whole on existing public image text data sets. The data sets we use are from the ICDAR 2003 Robust Reading challenge [3], Chars74K [18], and our SVT data set. In our experiments, we compare to results attained using conventional OCR systems ABBYY FineReader 9.0 and Tesseract OCR [9], referred to as ABBYY and TESS. In using the OCR engines, we experimented with pre-thresholding the images using the technique from [13], where they performed locally adaptive thresholding with a heuristic for a parameter sweep at each pixel. However, we found that deferring the thresholding task to the individual OCR engines resulted in better accuracy, and so we only report those results. In all our experiments, we resized images to take on a height of 50 pixels and used  $4 \times 4$  pixel cells with 10 orientation bins for the HOG features.

<sup>9</sup> <http://code.google.com/p/tesseract-ocr/>

## 6.1 Character Classification Results

We benchmarked our character classifier on the Chars74K-5, Chars74K-15, and ICDAR03-CH data sets. The Chars74K-5 and Chars74K-15 contained 5 and 15 training instances per class, respectively, while the test sets included the same 15 instances of each character class. The ICDAR03-CH data set is the character classification subproblem from the ICDAR Robust Reading data set. In all data sets, the characters included upper and lowercase letters, and digits 0 through 9; in total 62 symbols. Our evaluation methodology mirrored that of [18] and our results are reported next to theirs in Table 1.

In Table 1, our classifier is labeled as HOG+NN and is displayed in bold in the first row. The next three rows are reproduced from [18]. The first is Multiple Kernel Learning (MKL), which is a combination of a number of features described in [18]. In that work, results for MKL were only reported on the Chars74K-15, accounting for the dashes in the other two columns. The next two rows show performance using features from Geometric Blur (GB) [20] and Shape Context (SC) [21], and classified using Nearest-Neighbor (NN) as reported in [18]. The methods listed were the ones that performed best from [18].

**Table 1.** Results for character classification. Our HOG+NN approach performs best on the three benchmarks, demonstrating the benefit of using HOG features for character classification.

Feature	Chars74K-5	Chars74K-15	ICDAR03-CH
<b>HOG+NN</b>	<b>45.33 ± .99</b>	<b>57.5</b>	<b>51.5</b>
MKL	-	55.26	-
GB+NN	36.9 ± 1.0	47.09	27.81
SC+NN	26.1 ± 1.6	34.41	18.32
ABBYY	18.7	18.7	21.2
TESS	17.3	17.3	17.4

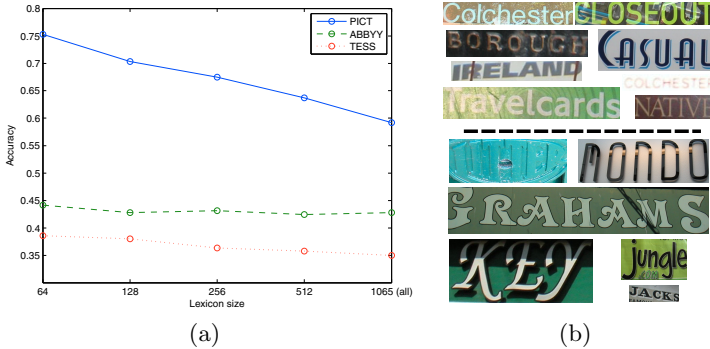
Our HOG+NN classifier outperforms those tested in [18] in all three benchmarks, and more significantly on the Chars74K-5 and ICDAR03-CH. However, we note that any suitable classification technique that can produce a list of discrete character detections can be substituted into the word recognition pipeline.

## 6.2 Word Recognition Results

We ran experiments on the ICDAR03-WORD and SVT-WORD data sets: the word recognition benchmarks of both data sets. Unlike SVT-WORD, ICDAR03-WORD is not explicitly structured for word spotting. Therefore, in our experiments, we construct lexicons synthetically using the ground truth. In both benchmarks, we use the exact same parameter settings and character training data, from ICDAR. In our comparisons to ABBYY and TESS, we provided the lexicons in the form of custom dictionaries and corrected OCR output to be the word with the smallest edit-distance in the lexicon.

**Table 2.** Number of trials for each lexicon size

Lexicon size	64	128	256	512	1065
Trials	16	8	4	2	1



**Fig. 5.** Subfigure (a) shows the performance of our method PICT, and OCR engines Abbyy FineReader 9.0 (ABBYY) and Tesseract OCR (TESS) on the ICDAR word benchmark. In this experiment, synthetic lexicons were created out of the ground truth in each run. We provided custom dictionaries to ABBYY and TESS and corrected their output to the nearest lexicon word by edit-distance. The y-axis marks word recognition accuracy and the x-axis marks the lexicon size. The full test size is 1,065 word images. In subfigure (b), the examples above the line are those that PICT only recognizes correctly, and the examples below are when all methods fail.

**ICDAR Robust Reading: Word Recognition.** In this experiment, we compare our approach, labeled as PICT, to the OCR engines ABBYY and TESS on ICDAR03-WORD. For simplicity, we filtered out words containing symbols other than letters and numbers, leaving 1,065 testing images. To formulate this problem as word spotting, we constructed tests of various sizes where we built synthetic lexicons out of the ground truth words for a particular test run. We divided the test set according to Table 2.

For each size  $k$ , we took all our testing data, randomized the order, and tested on contiguous chunks of size  $k$  until all of the test data was used. For example, when  $k = 64$ , we randomized the order of the test data and sampled sections of 64 images at a time (16 sections). We evaluated the three systems on each group of images where the lexicon consisted of words only from that set.

Figure 5 shows the word recognition results. The results are averaged over all the trials for each lexicon size. In our results, we see that at a lexicon size of 1,065, PICT significantly outperforms both OCR engines by over 15% and has more than 30% improvement when limiting the lexicon size to 64.

**Street View Text: Word Recognition.** In this benchmark, we tested ABBYY, TESS, and PICT on our Street View Text benchmark. On the SVT benchmark, PICT used the exact same training data and parameters as used in ICDAR03-WORD. No character training data from SVT was used. The test



**Fig. 6.** In our analysis, we use a simple and intuitive heuristic based on edge detection to group images into EASY and HARD. The EASY examples are typically those whose characters are well outlined, and the HARD ones typically contain more broken characters and edges from the background and shadows. This is a coarse estimate of those images that are more CAPTCHA-like.

size was 514 word images and each image had an associated list of businesses to categorize from. The accuracies for TESS, ABBYY, and PICT were 31.5%, 47.7%, and 59.0% respectively. Our PICT approach shows significant improvement over the OCR engines.

*Implementation Details:* The system was implemented in C++ using the OpenCV framework. Average processing time to run PICT was under six seconds on an Intel Core 2 processor.

## 7 Error Analysis

In an attempt to better understand the complexity of image text as it relates to the performance of conventional OCR, we introduce a simple diagnostic to gauge image difficulty. In both ICDAR and SVT data sets, there are examples of words that span the difficulty spectrum: some are well-suited for OCR while others present a challenge approaching that of a CAPTCHA. In our analysis, we separate the data into two groups, ‘EASY’ and ‘HARD’, based on a simple heuristic that is independent of either OCR engine. The intuition behind our heuristic is that easy examples are likely to have continuous edges around each character and few spurious edges from the background. We ran a Canny edge detector [22] on the the data and separated the images by calculating the number of continuous edges divided by the image’s aspect ratio. This value represents approximately the number of line segments in a space typically occupied by one to two characters. We placed images with values between 1 and 3.5 into the EASY category, and all others into the HARD category; see Figure 6 for examples of each category. In the EASY category, we can see that the edges around characters are often reliably traced, whereas in the HARD category, many edges are picked up from the background and shadows. Table 3 shows the breakdown of results after separating the data.

While this is not meant to be a definitive method for categorizing the data – indeed, there could be a more sophisticated heuristic to accurately identify

**Table 3.** This table shows the breakdown of results after applying our image diagnostic to categorize images as EASY and HARD. The proportion of the easy data for ICDAR and SVT data sets were 40% and 33% respectively.

METHOD	ICDAR (1065)			SVT		
	ALL	EASY (40%)	HARD (60%)	ALL	EASY (33%)	HARD (67%)
TESS	35.0	41.7	30.5	31.5	43.2	25.8
ABYY	42.8	56.9	33.4	47.7	62.7	40.3
<b>PICT</b>	<b>59.2</b>	<b>65.0</b>	<b>55.3</b>	<b>59.0</b>	<b>63.9</b>	<b>56.8</b>

**Table 4.** This table shows the breakdown of how often the two OCR engines determine the that image *does not contain readable text*. This situation constitutes a large portion of the overall errors in each engine.

METHOD	ICDAR (1065)			SVT		
	ALL	EASY (40%)	HARD (60%)	ALL	EASY (33%)	HARD (67%)
TESS	33.8	32.6	34.6	46.5	42.0	48.4
ABYY	45.2	34.5	52.4	44.6	29.6	51.9



**Fig. 7.** This figure shows some advantages of using part based object detection. In the images of ‘MARLBORO’ and ‘STUFF’, character segmentation is extremely challenging because of the cast shadows and letter designs. Using the character detection approach allows us to avoid explicit segmentation and instead relies on local peaks from our character detector. The configuration of the word ‘Marriott’ shows how a pictorial structure model is tolerant of minor errors in the part detections. We can see that even though the first ‘r’ is not in the correct position, the total configuration cost for the word is better than that of the others associated with that image.

text that can be read at scanned document levels – it is a simple and intuitive measure of image text complexity and provides a coarse estimate of how difficult an image of text is to segment. We can see all the methods perform significantly better on the EASY subset and the OCR methods suffer greater reductions on the HARD subset.

One reason for the significant performance drop of the OCR methods is that proper character segmentation is likely more challenging on the HARD set. The improvement in performance of the PICT model can be attributed to the fact that it avoids character segmentation, instead relying on character detection in a sliding window fashion. These detections are collected using a part based word

model designed that is robust to small errors. Figure 7 shows examples of these situations. In the images for ‘MARLBORO’ and ‘STUFF’, they are complex in appearance and suffer from cast shadows; as a result, accurate segmentation is extremely challenging. However, the detection approach focuses on finding local maxima in the response from the character classifier rather than segmentation. In the ‘Marriott’ example, a single misdetected part, the letter ‘r’, still results in word configuration score that allows it to be categorized correctly. While it is the case that minor errors in character classification are corrected using edit-distance for the OCR engines, we see from Table 4 that a common failure case is when the OCR engine returns no reading at all, suggesting that significant errors in segmentation can result in irrecoverable errors for OCR. The performance of PICT on the HARD subsets is what sets it apart from the OCR methods.

## 8 Conclusion

In this paper we explored the problem of word spotting and evaluated different methods to solve the problem. We have shown that approaching word spotting as a form of object recognition has the benefits of avoiding character segmentation – a common source of OCR errors – and is robust to small errors in character detection. When dealing with words in the wild, it is often the case that accurate segmentation is unattainable, and especially in these cases, our detection based approach shows significant improvement. While there is still room for improvement in performance, we have shown that framing the word spotting problem as generic object recognition is a promising new direction.

**Acknowledgments.** We thank Boris Babenko and Steve Branson for helpful conversations, and Grant Van Horn for assistance with data collection. This material is based upon work supported by NSF CAREER Grant No. 0448615, an NSF Graduate Research Fellowship, a Google Research Award, and the Amazon AWS in Education Program.

## References

1. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: Captcha: Using hard AI problems for security. In: Eurocrypt (2003)
2. 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)
3. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: ICDAR 2003 robust reading competitions. In: ICDAR (2003)
4. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI* 22, 1349–1380 (2000)
5. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: CVPR (2003)
6. Nagy, G.: At the frontiers of OCR. *Proceedings of IEEE* 80, 1093–1100 (1992)



7. Mori, S., Suen, C.Y., Yamamoto, K.: Historical review of OCR research and development. *Document Image Analysis*, 244–273 (1995)
8. Casey, R.G., Lecolinet, E.: A survey of methods and strategies in character segmentation. *IEEE Trans. PAMI* 18, 690–706 (1996)
9. Chellapilla, K., Larson, K., Simard, P.Y., Czerwinski, M.: Designing human friendly human interaction proofs (HIPs). In: *CHI* (2005)
10. Wu, V., Manmatha, R., Riseman, E.M.: Textfinder: An automatic system to detect and recognize text in images. *IEEE Trans. PAMI* 21, 1224–1229 (1999)
11. Sato, T., Kanade, T., Hughes, E.K., Smith, M.A., Satoh, S.: Video OCR: indexing digital new libraries by recognition of superimposed captions. *Multimedia Systems* 7, 385–395 (1999)
12. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Trans. PAMI* 31, 1733–1746 (2009)
13. Chen, X., Yuille, A.L.: Detecting and reading text in natural scenes. In: *CVPR* (2004)
14. Vanhoucke, V., Gokturk, S.B.: Reading text in consumer digital photographs. In: *SPIE* (2007)
15. Mori, G., Malik, J.: Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In: *CVPR* (2003)
16. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. *IEEE Trans. on Computers* 22, 67–92 (1973)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. *IJCV* 61, 55–79 (2005)
18. de Campos, T., Babu, B., Varma, M.: Character recognition in natural images. In: *VISAPP* (2009)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR* (2005)
20. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: *CVPR* (2005)
21. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI* 24, 509–522 (2002)
22. Canny, J.: A computational approach to edge detection. *IEEE Trans. PAMI* 8, 679–698 (1986)

# A Stochastic Graph Evolution Framework for Robust Multi-target Tracking\*

Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K. Roy-Chowdhury

Dept. of Electrical Engineering, University of California, Riverside, CA 92521, USA

**Abstract.** Maintaining the stability of tracks on multiple targets in video over extended time periods remains a challenging problem. A few methods which have recently shown encouraging results in this direction rely on learning context models or the availability of training data. However, this may not be feasible in many application scenarios. Moreover, tracking methods should be able to work across different scenarios (e.g. multiple resolutions of the video) making such context models hard to obtain. In this paper, we consider the problem of long-term tracking in video in application domains where context information is not available a priori, nor can it be learned online. We build our solution on the hypothesis that most existing trackers can obtain reasonable short-term tracks (tracklets). By analyzing the statistical properties of these tracklets, we develop associations between them so as to come up with longer tracks. This is achieved through a stochastic graph evolution step that considers the statistical properties of individual tracklets, as well as the statistics of the targets along each proposed long-term track. On multiple real-life video sequences spanning low and high resolution data, we show the ability to accurately track over extended time periods (results are shown on many minutes of continuous video).

## 1 Introduction

Multiple object tracking is the most fundamental task for higher level automated video content analysis. Although a large number of trackers exist, stable, long-term tracking is still a challenging problem. Common reasons which cause tracking failure are occlusion, illumination change, clutter and sensor noise. Moreover, for multiple targets, we have to consider the interaction between the targets which may cause errors like switching between tracks, missed detections and false detections. Therefore, detection and correction of the errors in the tracks is the key to robust long term tracking.

Many state-of-the-art tracking algorithms focus on how to avoid losing track. They usually rely on training data or learning context models (e.g. some recent papers like [11, 11, 16]). In many situations, there may not be enough data for training or learning context models. For example, videos downloaded from

---

\* This work was supported in part by NSF grant IIS-0712253 and subcontract from Mayachitra Inc., through a DARPA STTR award (#W31P4Q-08-C-0464).

Youtube are usually a few minutes in length and from a variety of contexts. Analysis of these videos requires tracking and there is no separate data available to learn models.

In this paper, we consider the problem of long-term tracking in video in application domains where context information is not available a priori, nor can it be learned online. *We are not proposing our method as an alternative to learning models, rather as an approach for applications where such data is not available.* Building on the hypothesis that most existing trackers can obtain reasonable short-term tracks (tracklets), we propose a stochastic graph evolution framework to understand the association between tracklets so as to come up with longer tracks by analyzing the statistical properties of individual tracklets, as well as the statistics of the targets along each proposed long-term track.

Our approach is original in the following ways.

- We come up with a measure of the accuracy of the tracking, so that we can determine when the tracking error is increasing and identify the tracklets.
- We propose a prediction-based affinity modeling approach by searching for optimal associations in the target feature space using a stochastic sampling method. We show that this provides higher accuracy as opposed to heuristically selecting a fixed affinity model. This process leads to a weighted graph with the tracklets as nodes and affinity scores as weights.
- We consider long-term interdependencies between the target tracklet features and use it to correct for wrong correspondences. This is achieved by evolving the graph weights through a stochastic sampling approach. The underlying hypothesis for this step is that along a correct track the variation of the target features will be lower than along a wrong track.

Through this process, we are able to get stable long-term tracks of multiple targets without the need for extra training data. Our method analyzes the video in a time-window (maximum duration of a few minutes) in a batch process; thus there is a delay in the analysis, which is often a non-issue in many applications.

## 1.1 Related Work

To track multiple objects, a lot of effort has been devoted to making data association based on the results of object detection. Multi-Hypothesis Tracking (MHT) [13] and Joint Probabilistic Data Association Filters (JPDAF) [2] are two representative methods. In order to overcome the large computational cost of MHT and JPDAF, various optimization algorithms such as Linear Programming [9], Quadratic Boolean Programming [10], and Hungarian algorithm [12] are used for data association. In [17], data association was achieved through a MCMC sampling based framework. These methods rely on the precision of object detection, which can not be guaranteed in complex scenarios. On the other hand, some static tracking methods (e.g. Kalman filter and particle filter [8]) and kernel tracking algorithm (e.g. mean-shift tracker [3]) release the requirement for object detection in every frame, but they are not powerful for tracking

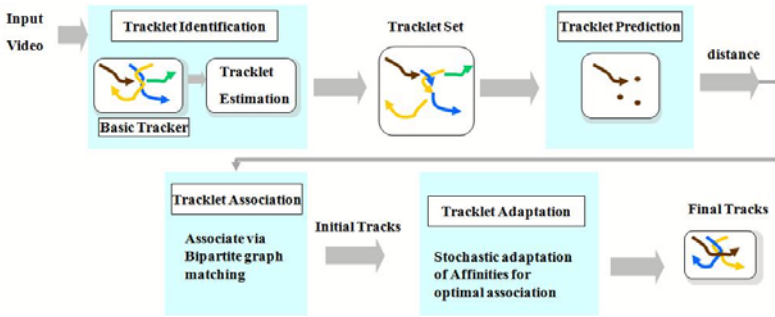


Fig. 1. Overview of proposed approach

multiple objects by themselves. In [7], particle filters were used to track multiple objects by incorporating probabilistic MHT for data association.

Many state-of-the-art tracking algorithms focus on how to avoid errors in tracking. In [18], the authors proposed a min-cost flow framework for global optimal data association. A tracklet association based tracking method was presented in [5], which fixed the affinity model heuristically and focused on searching for optimal associations. A HybridBoosted affinity model was learned in [11]. The method is built on the availability of training data under a similar environment, which may not be always feasible. The authors in [1] addressed the problem of learning an adaptive appearance model for object tracking. Context information was considered in [16] to help in tracking, by integrating a set of auxiliary objects which are learned online. Unfortunately, except for high resolution video, it is not easy to find these auxiliary objects.

We would like to clearly differentiate our approach with traditional Data Association Tracking (DAT) approaches which perform the tracking by detection instead of running a tracking algorithm. Unlike the DAT methods, our data association is done on the tracking results, not the detection result. Moreover, in most methods, there is very little attention paid on error recovery, i.e., if errors happen, how to detect and correct them. It is, however, at the heart of the proposed strategy.

## 2 Overview of Solution Strategy

Our system is initialized when new targets are detected. A basic tracker using particle filter is applied to generate the initial tracks. It can be replaced by any existing tracker, without affecting the other modules. However, errors cannot be avoided in the tracks generated by the basic trackers, especially in the presence of occlusions, disappearance of targets and close proximity of targets. In order to correct the errors, we propose a stochastic tracklet association and adaptation strategy.

Fig. 1 shows an overview of our long-term tracking system. We begin by identifying tracklets, i.e., the short-term fragments with low probability of error,

which are estimated from the initial tracks by evaluating the tracking performance. Details on estimation of tracklets are provided in Section 3.

The tracklets are then associated based on their affinities. Although an optimal affinity model could be learned [11], it requires the availability of training data. Instead of using a heuristically selected fixed affinity model, we propose a prediction based affinity modeling approach by searching for optimal predictions in the feature space based on Markov chain Monte Carlo (MCMC) sampling methods as detailed in Section 4. The tracklets are first extended in space and time through new predicted positions generated using the Metropolis Hastings algorithm. The affinity between two tracklets is modeled by the distance (in a suitable feature space) of the predicted ending of one tracklet to the starting of another. Using the affinity model, we create a tracklet association graph (TAG) with the tracklets as nodes and affinity scores as weights. The association of the tracklets can be found by computing the optimal paths in the graph. The optimal path computation is based on the principles of dynamic programming and gives the maximum a posteriori (MAP) estimate of tracklets' connections as the long-term tracks for each target. This is explained in Section 4.1.

The tracking problem could be solved optimally by the above tracklet association method if the affinity scores were known exactly and assumed to be independent. However, this can be a big assumption due to well known low-level image processing challenges, like poor lighting conditions or unexpected motion of the targets. The prediction based affinity model may not be enough to capture the variation. This leads us to develop a graph evolution scheme as described in Section 5. The affinities (i.e., the weights on the edges of TAG) are stochastically adapted by considering the distribution of the features along possible paths in the association graph to search for the global optimum. We design a loss function and an efficient optimization strategy for this process. The overall approach is able to track stably over minutes of video in challenging domains with no learning and context information.

### 3 Tracklet Identification

As mentioned earlier, we identify the tracklets from the initial tracks generated from the basic tracker. Then the problem of tracking over long-term video is equivalent to finding the best association between the tracklets. Note that although the particle filter based basic tracker is replaceable, it was chosen because the observation model is nonlinear and the posterior can temporarily become multimodal due to background clutter. We now describe our implementation of the basic tracker using a particle filter and the tracklet estimation scheme.

#### 3.1 Particle Filter Based Basic Tracker

**Initialization:** We use motion detection to automatically detect moving objects. The background modeling algorithm in [15] is used for its adaptability to illumination change, and to learn the multimodal background through time. Using

the learned background model, the moving objects can be detected. However, the background model may not be precise due to noise, which could produce false detections. By observing that most of our interested targets, like people and vehicles, are on ground plane, we estimate the rough ground plane area using the method proposed in [6]. Based on the ground plane information, false alarms can be removed significantly. We reiterate that this process is just one choice based on the current literature. It can be replaced and we do not assume that this step should work perfectly. In fact, the following stages are designed to correct for the errors here.

**System model:** The target regions are represented by rectangles with the state vector  $X_t = [x, y, \dot{x}, \dot{y}, l_x, l_y]$ , where  $(x, y)$  and  $(\dot{x}, \dot{y})$  are the position and velocity of a target in the  $x$  and  $y$  directions respectively, and  $(l_x, l_y)$  denote the size of the rectangle. We consider a linear dynamic model:  $X_t = AX_{t-1} + n_t$ ,

where  $A$  defines the deterministic system model and  $n_t$  is zero mean white Gaussian noise ( $n_t \sim \mathcal{N}(0, \Sigma_t)$ ).

**Observation model:** The observation process is defined by the likelihood distribution,  $p(I_t|X_t)$ , where  $X_t$  is the state vector and  $I_t$  is the image observation at  $t$ . Our observation models were generated by combining an appearance and a foreground response model, i.e.,

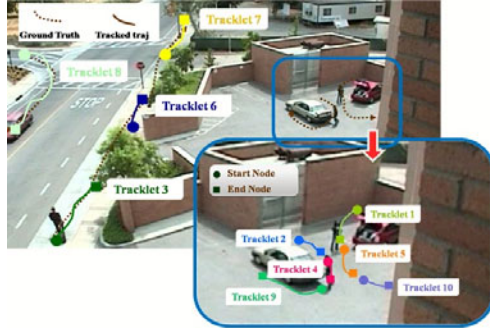
$$p(I_t|X_t) = p(I_t^a, I_t^f|X_t), \tag{1}$$

where  $I_t^a$  is the appearance information of  $I_t$  and  $I_t^f$  is the foreground response of  $I_t$  using the learned background model as described above.  $I_t^f$  is a binary image with “1” for foreground and “0” for background.

It is reasonable to assume that  $I_t^a$  and  $I_t^f$  are independent and thus (1) becomes  $p(I_t|X_t) = p(I_t^a|X_t)p(I_t^f|X_t)$ . The appearance observation likelihood is defined as  $p(I_t^a|X_t) \propto \exp\{-B(ch(X_t), ch_0)^2\}$ , where  $ch(X_t)$  is the color histogram associated with the rectangle region of  $X_t$  and  $ch_0$  is color histogram of the initialized target.  $B(\cdot)$  is the Bhattachayya distance between two color histograms. The foreground response observation likelihood is  $p(I_t^f|X_t) \propto \exp\{-(1 - \frac{\#F(X_t)}{\#X_t})^2\}$ , where  $\#F(X_t)$  is the number of foreground pixels in the rectangular region of  $X_t$  and  $\#X_t$  is the total number of pixels in that rectangle.  $\frac{\#F(X_t)}{\#X_t}$  represents the percentage of the foreground in that rectangle. The observation likelihood would be higher if more pixels in the rectangular region of  $X_t$  belong to the foreground.

### 3.2 Tracklet Estimation

Errors cannot be avoided in the tracks generated by any basic tracker. There are two common errors: lost track (when the track is no longer on any target, but on the background) and track switching (when targets are close and the tracks are on the wrong target). This leads us to the rules for tracklet estimation. We estimate when these errors happen and identify their spatio-temporal location, leading to the tracklets. An example is shown in Fig. 2.



**Fig. 2.** An example of tracklet identification. The ground truth trajectories are represented by brown dotted lines. The estimated tracklets due to detection of a lost track (track of the person in lower left corner due to occlusion) and targets' close proximity (the persons moving around the cars) are clearly shown in different colors.

**Detection of lost track:** The tracking error (TE) [2] or prediction error is the distance between the current observation and its prediction based on past observations. TE will increase when the tracker loses track and can be used to detect the unreliability of the track result. In our observation model, TE of tracked target  $\hat{X}_t$  is calculated by

$$TE(\hat{X}_t, I_t) = TE_a(\hat{X}_t, I_t) + TE_f(\hat{X}_t, I_t), \quad (2)$$

$$\text{where } TE_a(\hat{X}_t, I_t) = B(ch(X_t), ch_0)^2 \quad \text{and} \quad TE_f(\hat{X}_t, I_t) = \left(1 - \frac{\#F(X_t)}{\#X_t}\right)^2.$$

If a lost track is detected, it means the tracking result after this point is not reliable; in the tracking procedure, we stop doing tracking after this point and identify a tracklet. In the case of false detection (i.e., the detected target is a part of background), or target passes through a region with similar color, or a target stops, the background modeling algorithm will adapt to treat this as a part of the background, and thus  $TE_f$  will eventually increase. Then a lost track will be detected.

**Track Switching:** When targets are close to each other, a track switch can happen with high probability especially if the appearances of targets are similar. Thus, we inspect the distances between targets, and break the tracks into tracklets at the points where targets are getting close, as shown in Fig. 2.

## 4 Prediction Based Tracklet Affinity Modeling

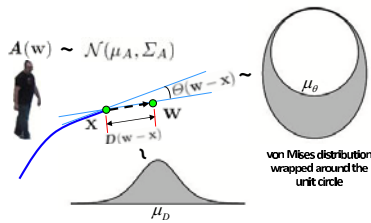
As mentioned in [11], in most previous work, simple affinity models are used by heuristically selecting parameters. The approach in [11] is able to automatically select among features and corresponding non-parametric models based on training data. However, without the availability of training data, searching in such an

affinity function space is not trivial. Under this condition, rather than directly search in the affinity function space, we propose a prediction based affinity modeling approach by searching for optimal predictions in the feature space based on MCMC sampling methods and using the predicted features to come up with the affinity measurements. This provides more robustness compared to using a fixed affinity measure, as shown in Table 3 in Section 6.

### 4.1 Tracklet Prediction and Association

The tracklet occurring earlier in time is referred to as the base-tracklet, while a tracklet beginning after the base-tracklet ended is referred to as the target-tracklet. In order to measure tracklet affinity, the base-tracklet is extended in the image motion/appearance-space  $M$  steps, where  $M$  could represent the number of frames that separate the end of the base-tracklet from the beginning of the target-tracklet or a fixed number of pre-determined steps. In order to choose new points for the base-tracklet, a form of MCMC called the Metropolis Hastings Algorithm is used to generate chains of random samples.

MCMC is a versatile tool for generating random samples that can be used in determining statistical estimates. By using this sampling method, the algorithm is able to take advantage of the base object’s motion and appearance information while also considering its relationship to the target-tracklet via the target distribution  $p_{tl}(\mathbf{z})$ . The target distribution relates points surrounding the starting point of the target-track to a probability measure. MCMC has the advantage of not requiring perfect knowledge of the target distribution  $p_{tl}(\mathbf{z})$  – it is enough to be able to evaluate it at a particular point, but not sample from it.



**Fig. 3.** An illustration of proposing a new point based on the proposal distribution

**The Proposal Distribution:** The proposal distribution  $q_{tl}(\mathbf{y}|\mathbf{z})$  allows us to generate samples from a distribution that is easy to sample from. Our proposal distribution was based on a combination of motion and appearance of each target. The direction of motion of each target is modeled using the von Mises distribution. The von Mises distribution has close ties to the normal distribution, however it is limited to angles about the unit circle as shown in Fig. 3. The pdf for the von Mises distribution takes the following form:

$$v(\theta|\mu_\theta, \kappa) = \frac{e^{\kappa \cos(\theta - \mu_\theta)}}{2\pi I_0(\kappa)}. \tag{3}$$

Here,  $I_0(\cdot)$  is the modified Bessel function of order zero. The parameters  $\mu_\theta$  and  $\kappa$  correspond to mean and variance in a normal distribution, which are learned within each base tracklet.

The speed of each target is modeled with a Normal distribution  $\mathcal{N}(\mu_D, \sigma_D)$ , where the mean  $\mu_D$  and variance  $\sigma_D$  are learned within each base tracklet. The



appearance model is described using a normal distribution on the color histogram of each target as  $\mathcal{N}(\mu_A, \Sigma_A)$ , where the parameters are also learned within each base tracklet.

So our proposal distribution is

$$q_{tl}(\mathbf{w}|\mathbf{x}) \propto v(\Theta(\mathbf{w} - \mathbf{x})|\mu_\theta, \kappa)\mathcal{N}(D(\mathbf{w} - \mathbf{x})|\mu_D, \sigma_D)\mathcal{N}(A(\mathbf{w})|\mu_A, \Sigma_A), \quad (4)$$

where  $\Theta(\mathbf{w} - \mathbf{x})$  and  $D(\mathbf{w} - \mathbf{x})$  represent the angle and distance between the proposed point  $\mathbf{w}$  and the end point of tracklet  $\mathbf{x}$  respectively, and  $A(\mathbf{w})$  represents the color histogram of proposed point  $\mathbf{w}$ . A new point is proposed by randomly producing motion direction, speed and appearance vector as shown in Fig. 3.

**The Target Distribution:** Proposed points from the base-tracklet were related to the starting point of the target-tracklet through the target distribution. The target distribution,  $p_{tl}(\mathbf{z})$ , was chosen as

$$p_{tl}(\mathbf{z}) \propto e^{-d_z}, \quad (5)$$

where  $d_z = \sqrt{d_a^2 + d_m^2}$  is a Euclidean combination of the normalized distance in the motion-space,  $d_m$ , and the Bhattacharyya distance,  $d_a$ , between the image histograms of the average base and target appearances.

**M-H Algorithm:** Given the proposal distribution,  $q_{tl}(\mathbf{w}|\mathbf{x})$ , where  $\mathbf{w}$  was the proposed point and  $\mathbf{x}$  was the last point in the tracklet and the target distribution  $p_{tl}(\mathbf{w})$ , the probability that a point was accepted was given as,

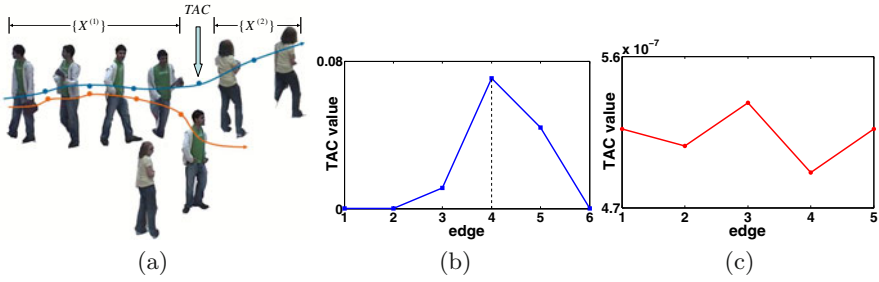
$$\rho_{tl}(\mathbf{x}, \mathbf{w}) = \min \left\{ \frac{p_{tl}(\mathbf{w})q_{tl}(\mathbf{x}|\mathbf{w})}{p_{tl}(\mathbf{x})q_{tl}(\mathbf{w}|\mathbf{x})}, 1 \right\}. \quad (6)$$

This process results in a sequence of accepted points for  $M$  time steps. The affinity between a base and target tracklet is computed as the distance  $d_z$  in (5) between the end of the predicted extension of the base tracklet and the beginning of the target tracklet.

**Tracklet Association** We can now define a Tracklet Association Graph where the nodes are the identified tracklets and the weights on the edges are the affinity scores. By splitting the beginning and end of each tracklet into two subsets, the problem of the tracklet association can be formulated as a maximum matching problem in a weighted bipartite graph. In this paper, we use the Hungarian algorithm [12] to find the maximum matching.

## 5 Tracklet Adaptation

If the affinity scores (edge weights) of the bipartite graph were known exactly and assumed to be independent, the tracking problem could be solved optimally by the tracklet association method described above. However, it is not uncommon for some of the similarities to be estimated wrongly since they depend on detected features which is not a perfect process. As we show in Fig. 4, if the similarity



**Fig. 4.** (a) Tracklets of two targets obtained from Videoweb courtyard dataset of Section 6 ground truth track of the person in green T-shirt is shown with orange line, and the association results before adaptation are shown with blue line. (b)-(c): TAC values along the incorrect and correct association results respectively, (note that the range of the y-axis in (c) is much smaller than (b)). It is clear that TAC has a peak at the wrong link; thus the variance of TAC along the wrongly associated tracklets is higher than the correct one.

estimation is incorrect for one pair of tracklets, the overall inferred long track may be wrong even if all the other tracklets are connected correctly.

We address this issue by constructing a graph evolution strategy, in which the weights (i.e., affinity scores) on the edges of the tracklet association graph are adapted by measuring the similarity of observed features along a path that is generated after tracklet association. We adopt the affinity adaptation method proposed in [14], but instead of adapting deterministically which may be stuck at a local optimum, we propose a Metropolis-Hastings based adaptation scheme with the potential to reach the global optimal.

### 5.1 Tracklet Association Cost Function

To model the spatio-temporal variation of the observed features along a path, a Tracklet Association Cost (TAC) is defined motivated by [14]. Given an estimated track for the  $q^{th}$  target,  $\lambda_q$ , TAC is defined on each edge  $e_{ij} \in \lambda_q$ . The feature vector of the tracklets before (in time)  $e_{ij}$  on  $\lambda_q$  and those after  $e_{ij}$  are treated as two clusters. An illustration of TAC calculation is shown in Fig. 4(a).

Let  $\{X\}$  be the set of feature (e.g., appearance) of all  $N$  tracklets along the path and let them be clustered into  $\{X^{(1)}\}$  and  $\{X^{(2)}\}$  with respect to each edge  $e_{ij} \in \lambda_q$ . Let the mean  $m$  of the features in  $\{X\}$  be  $m = \frac{1}{N} \sum_{x \in \{X\}} x$ . Let  $m_i$  be the mean of  $N_i$  data points of class  $\{X^{(i)}\}$ ,  $i = 1, 2$ , such that  $m_i = \frac{1}{N_i} \sum_{x \in \{X^{(i)}\}} x$ . Let  $S_T$  be the variance of the all observed feature  $x$  along the path, i.e.,  $S_T = \sum_{x \in \{X\}} |x - m|^2$  and  $S_W$  be the sum of the variances along each sub-path,  $\{X^{(1)}\}$  and  $\{X^{(2)}\}$ , i.e.,  $S_W = \sum_{i=1}^2 S_i = \sum_{i=1}^2 \sum_{x \in \{X^{(i)}\}} |x - m_i|^2$ .

The TAC for  $e_{ij}$  is defined as

$$TAC(e_{ij}) = \frac{|S_T - S_W|}{|S_W|} \triangleq \frac{|S_B|}{|S_W|}. \tag{7}$$

Thus the TAC is defined from Fisher's linear discriminant function [4] and measures the ratio of the distance between different clusters,  $S_B$ , over the distances between the members within each cluster  $S_W$ . If all the feature nodes along a path belong to the same target, the value of TAC at each edge  $e_{ij} \in \lambda_q$  should be low, and thus the variance of TAC over all the edges along the path should also be low. If the feature nodes belonging to different people are connected wrongly, we will get a higher value of TAC at the wrong link, and the variance of TAC along the path will be higher. Thus, the distribution of TAC along a path can be used to detect if there is a wrong connection along that path.

We can now design a loss function for determining the final tracks by analyzing features along a path. We specify the function in terms of the Tracklet Association Cost (TAC) function. Thus, we adapt the affinity scores to minimize

$$L(\lambda_q) = \sum_{\lambda_q} \text{Var}(TAC(e_{ij} \in \lambda_q^{(n)})). \quad (8)$$

## 5.2 Metropolis-Hastings Based Adaptation of Tracklet Association

Whenever there is a peak<sup>1</sup> in the TAC function for some edge along a path, the validity of the connections between the features along that path is under doubt. As per the Metropolis-Hastings method, we will propose a new candidate affinity score  $s'_{ij}$  on this edge where the peak occurs using a proposal distribution  $q_{af}(s'_{ij}|s_{ij})$ , where  $s_{ij}$  is the affinity score on edge  $e_{ij}$ . The proposal distribution  $q_{af}(s'_{ij}|s_{ij})$  is chosen to be a uniform distribution of width  $2\delta$ , i.e.,  $U(s_{ij} - \delta, s_{ij} + \delta)$ , since without additional information, uniform distribution can be a reasonable guess of the new weights. Any other distribution can be chosen based on the application.

We then recalculate the maximum matching paths,  $\lambda'_q$ , of the new feature graph. The target probability  $p_{af}(\cdot)$  is defined as  $p_{af}(s_{ij}) \propto \exp(-L(\lambda_q))$ , and  $p_{af}(s'_{ij}) \propto \exp(-L(\lambda'_q))$ . The candidate weight  $s'_{ij}$  is accepted with probability  $\rho_{af}(s_{ij}, s'_{ij})$  as

$$\rho_{af}(s_{ij}, s'_{ij}) = \min \left\{ \frac{p_{af}(s'_{ij})q_{af}(s_{ij}|s'_{ij})}{p_{af}(s_{ij})q_{af}(s'_{ij}|s_{ij})}, 1 \right\}. \quad (9)$$

Our adaptation scheme is summarized below.

1. Construct a weighted graph  $G = (V, E, S)$ , where the vertices are the tracklets and edge weights are set as described in Section 4.
2. Estimate the optimal paths,  $\tilde{\lambda}_q$  based on bipartite graph matching.
3. Compute the TAC for each  $e_{ij} \in \tilde{\lambda}_q$ .
4. Propose a weight  $s'_{ij}$  on the link where the TAC peak occurs based on a proposal distribution.

<sup>1</sup> The peak is detected if it is above a threshold, which is defined as  $E\{TAC(e_{ij} \in \lambda_q)\} + 2\sqrt{\text{Var}(TAC(e_{ij} \in \lambda_q))}$ .

**Table 1.** Evaluation metrics

Name	Definition
GT	Num of ground truth trajectories
MT%	Mostly tracked: Percentage of GT trajectories which are covered by tracker output more than 80% in time
ML%	Mostly lost: Percentage of GT trajectories which are covered by tracker output less than 20% in time
FG	Fragments: The total Num of times that the ID of a target changed along a GT trajectory
IDS	ID switches: The total Num of times that a tracked target changes its ID with another target
RS%	Recover from short term occlusion
RL%	Recover from long term occlusion

- Recalculate the maximum matching paths,  $\lambda'_q$ , of the new feature graph. We accept the new graph with probability  $\rho_{af}(s_{ij}, s'_{ij})$  in (9).
- Repeat Steps 4 and 5 until either a predefined iteration number is reached or the system reaches some predefined stopping criterion.

## 6 Experimental Results

To evaluate the performance of our system, we show results on two different data sets. The CAVIAR (<http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1>) is captured in a shopping mall corridor with heavy inter-object occlusion. The Videoweb dataset (<http://vwdata.ee.ucr.edu>) is a wide area multi-camera dataset consisting of low and high resolution videos. We consider two subsets of videos. The first is an outdoor low resolution parking lot scene, and the second is a relatively high resolution courtyard scene with intensive occlusion and clutter.

To evaluate the performance of our system quantitatively, we adopt the evaluation metrics for tracking defined in [11] and [18]. In addition, we define RS and RL to evaluate the ability of recovering from occlusion (see Table 1). Although we show results on datasets that others have worked with, it should be noted that we are not proposing our method as an alternative to those that use/learn context models, rather as an approach to be used when such models are not available. Therefore, our results should be analyzed with the ground truth, rather than against those that rely on such knowledge.

**Results on CAVIAR dataset:** In CAVIAR dataset, the inter-object occlusion is high and includes long term partial occlusion and full occlusion. Moreover, frequent interactions between targets such as multiple people talking and walking in a group make tracking more challenging. We show our results on the relatively more challenging part of the dataset which contains 7 videos (TwoEnterShop3, TwoEnterShop2, ThreePastShop2, ThreePastShop1, TwoEnterShop1, OneShopOneWait1, OneStopMoveEnter1)<sup>2</sup>. Table 2 shows the comparison among the proposed method, the min-cost flow approach in [18], HybridBoosted affinity

<sup>2</sup> Compared with other sequences in CAVIAR (e.g. TwoLeaveShop2, OneStopNoEnter1 and OneStopMoveNoEnter1), the challenge of the set we test on is obvious.

**Table 2.** Tracking Results on CAVIAR data set. Results of [11] and [18] are reported on 20 sequences; basic particle filter and proposed method are reported on 7 most challenging sequences of the dataset. Our test data has totally 12308 frames for about 500 sec.

	GT	MT	ML	FG	IDS	RS	RL
Zhang <i>et al.</i> [18]	140	85.7%	3.6%	20	15	-	-
Li <i>et al.</i> [11]	143	84.6%	1.4%	17	11	-	-
Basic particle filter	75	53.3%	10.7%	15	19	18/42	0/8
Proposed method	75	84.0%	4.0%	6	8	36/42	6/8

**Table 3.** Tracking Results on one sequence of CAVIAR dataset. Proposed approach is a combination of basic particle filter, prediction based affinity model and track adaptation.

	GT	MT	ML	FG	IDS	RS	RL
Basic particle filter	18	44.4%	22.2%	7	6	4/14	0/5
Simple Affinity model	18	66.6%	5.6%	2	4	12/14	2/5
Prediction-Based Affinity model	18	72.2%	0.0%	2	3	13/14	3/5
Proposed method	18	83.3%	0.0%	2	1	13/14	4/5

modeling approach in [11] and a basic particle filter. The results in [11,18] are reported on 20 sequences in CAVIAR. It can be seen that our method achieves similar performance as in [11,18]. It should also be noted that [11,18] are built on the availability of training data under similar environment (e.g. other 6 sequences in CAVIAR are used for training in [18]), while our method does not rely on any training; also our results are for the most challenging sub-part of the dataset. Some sample frames with results are shown in Fig. 5 (a). In the supplementary material, we show results on continuously tracking this data.

In order to show the achievement of each step (i.e., the prediction based affinity modeling and tracklet adaptation) of our proposed method, we compare the performances of the basic particle filter, a simple affinity model followed by bipartite graph match, prediction based affinity model without tracklet adaptation step, and the complete proposed approach on one of the sequences (the one shown in the supplementary material). The simple affinity model is constructed by directly using the average angle and speed of motion and average color histogram similar to [18]. It is clearly shown in Table 3 that our method has much less Fragments (FG) and ID Switches (IDS) and the adaptation part can further correct the wrong connections.

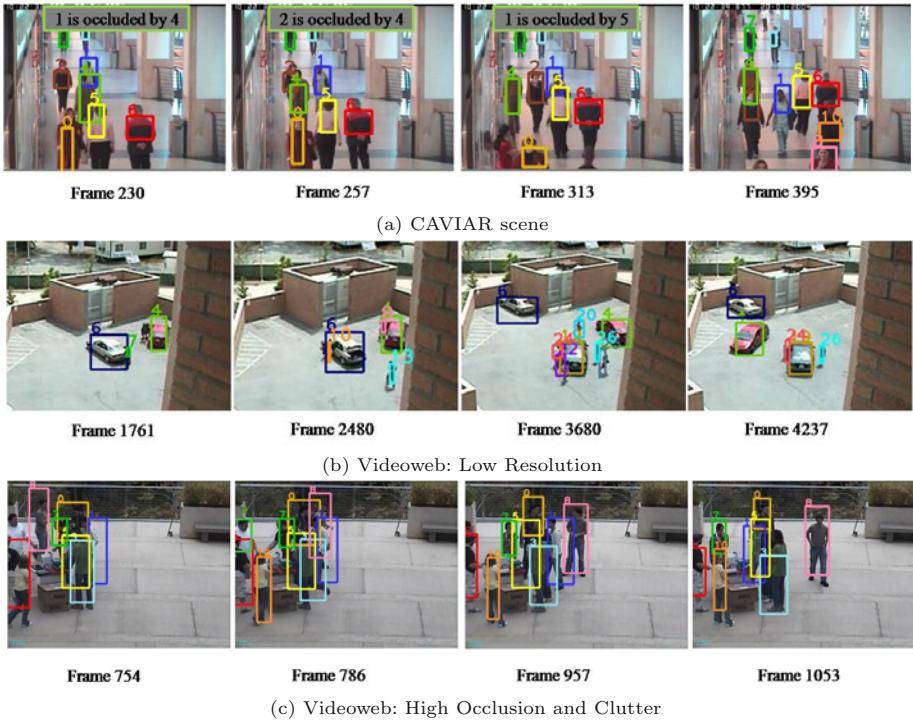
**Results on Videoweb dataset – Low-resolution Example:** The first part of Videoweb dataset we use is a low resolution parking lot scene. The target categories include people, cars and motorcycle (any object which is below 15 pixels in width is not taken into account). The low resolution makes tracking more challenging, especially in outdoor scenes since the illumination is always unstable and the appearance is hard to extract. The results of our methods are shown in Table 4. Some sample frames and tracking results are shown in 5 (b).

**Table 4.** Tracking Results on parking lot scene of Videoweb dataset. 4 sequences of totally 14673 frames (980 sec.) were used.

	GT	MT	ML	FG	IDS	RS	RL
Basic particle filter	90	80%	10.0%	20	6	5/19	1/8
Proposed method	90	90%	4.4%	8	3	15/19	5/8

**Table 5.** Tracking Results on courtyard scene of Videoweb dataset, 4 sequences of totally 8254 frames (550 sec.) were used.

	GT	MT	ML	FG	IDS	RS	RL
Basic particle filter	48	41.7%	14.6%	9	17	10/35	2/15
Proposed method	48	66.7%	6.25%	5	8	29/35	12/15



**Fig. 5.** (a): Tracking results on CAVIAR dataset. (b): Tracking results on Videoweb dataset - low resolution parking lot scene. (c): Tracking results on Videoweb dataset - high clutter and occlusion courtyard scene.

**Results on Videoweb dataset – High Occlusion and Clutter Example:** The second part of Videoweb dataset consists of multiple people interacting in a courtyard. It is almost impossible to track with a basic tracker because of very

high occlusion. Also, an adaptive background model is hard to build for this level of occlusion. The tracking result shows our method using the proposed strategy can get reasonable results even at this level of occlusion. The performance on this dataset is shown in Table 5. Some sample frames with tracking results are shown in 5 (c). Results on tracking about 45 seconds of this scene are shown in the supplementary material<sup>3</sup>

## 7 Conclusions

In this paper, we considered the problem of long-term tracking in video in application domains where context information is not available a priori, nor can it be learned online. We built our solution on the hypothesis that most existing trackers can obtain reasonable short-term tracks (tracklets). We then developed associations between them so as to come up with longer tracks. Finally, we proposed a graph evolution method to search for optimal association, then providing robustness to inaccuracies in feature similarity estimation. Promising results are shown on challenging data sets.

## References

1. Babenko, B., Yang, M., Belongie, S.: Visual Tracking with Online Multiple Instance Learning. In: IEEE CVPR (2009)
2. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press, London (1988)
3. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence (May 2003)
4. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley Interscience, Hoboken (2001)
5. Ge, W., Collins, R.: Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In: British Machine Vision Conference (2008)
6. Hoiem, D., Efros, A., Hebert, M.: Geometric Context from a Single Image. In: IEEE ICCV (2005)
7. Hue, C., Cadre, J.L., Prez, P.: Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion. IEEE Trans. on Signal Processing (2002)
8. Isard, M., Blake, A.: Condensation - Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision (1998)
9. Jiang, H., Fels, S., Little, J.: A Linear Programming Approach for Multiple Object Tracking. In: IEEE CVPR (2007)
10. Leibe, B., Schindler, K., Gool, L.V.: Coupled Detection and Trajectory Estimation for Multi-Object Tracking. In: IEEE ICCV (2007)
11. Li, Y., Huang, C., Nevatia, R.: Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene. In: IEEE CVPR (2009)
12. Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W.: Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions. In: IEEE CVPR (2006)

---

<sup>3</sup> More extensive tracking results are available on the author's webpage.

13. Reid, D.: An Algorithm for Tracking Multiple Targets. *IEEE Trans. Automatic Control* 24(6), 843–854 (1979)
14. Song, B., Roy-Chowdhury, A.: Stochastic Adaptive Tracking in a Camera Network. In: *IEEE ICCV* (2007)
15. Stauffer, C., Grimson, W.: Adaptive Background Mixture Models for Real-time Tracking. In: *IEEE CVPR* (1998)
16. Yang, M., Wu, Y., Hua, G.: Context-Aware Visual Tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (July 2009)
17. Yu, Q., Medioni, G., Cohen, I.: Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. In: *IEEE CVPR* (2007)
18. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: *IEEE CVPR* (2008)



# Backprojection Revisited: Scalable Multi-view Object Detection and Similarity Metrics for Detections

Nima Razavi<sup>1</sup>, Juergen Gall<sup>1</sup>, and Luc Van Gool<sup>1,2</sup>

<sup>1</sup> Computer Vision Laboratory, ETH Zurich

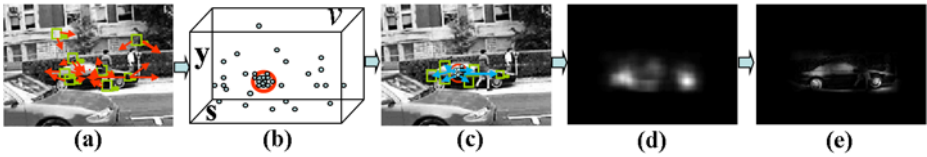
<sup>2</sup> ESAT-PSI/IBBT, KU Leuven

{nravazi,gall,vangool}@vision.ee.ethz.ch

**Abstract.** Hough transform based object detectors learn a mapping from the image domain to a Hough voting space. Within this space, object hypotheses are formed by local maxima. The votes contributing to a hypothesis are called support. In this work, we investigate the use of the support and its backprojection to the image domain for multi-view object detection. To this end, we create a shared codebook with training and matching complexities independent of the number of quantized views. We show that since backprojection encodes enough information about the viewpoint all views can be handled together. In our experiments, we demonstrate that superior accuracy and efficiency can be achieved in comparison to the popular one-vs-the-rest detectors by treating views jointly especially with few training examples and no view annotations. Furthermore, we go beyond the detection case and based on the support we introduce a part-based similarity measure between two arbitrary detections which naturally takes spatial relationships of parts into account and is insensitive to partial occlusions. We also show that backprojection can be used to efficiently measure the similarity of a detection to all training examples. Finally, we demonstrate how these metrics can be used to estimate continuous object parameters like human pose and object’s viewpoint. In our experiment, we achieve state-of-the-art performance for view-classification on the PASCAL VOC’06 dataset.

## 1 Introduction

As an important extension of the Generalized Hough Transform (GHT) [2], the Implicit Shape Model (ISM) [3] trains a codebook of local appearance by clustering a training set of sparse image features and storing their relative location and scale with respect to the object center. For detection, sparse image features are extracted from a test image and are matched against the codebook casting probabilistic votes in the 3D Hough accumulator as illustrated in Fig. 1(a,b). The support is composed of all the votes that contribute to a detection. The backprojection of the support gives direct evidence of the object’s presence and can be used as hypothesis verification, Fig. 1(c,d). For instance, by providing pixel-accurate segmentation of the training data and storing this information in the codebook, the backprojection can be augmented by top-down segmentation [3].



**Fig. 1.** Object detection using Implicit Shape Models: (a) Features are matched against the codebook  $\mathcal{C}$  casting votes to the voting space  $\mathcal{V}$ . (b) The local maxima of the voting space is localized and the votes contributing to it are identified (inside the red circle). (c) The votes are backprojected to the image domain creating the backprojection mask. (d-e) Visualization of the backprojection mask. Note that the mask does not include the area occluded by a pedestrian. The image is taken from UIUC cars dataset [11].

Although the support and its backprojection have been used for verification and meta-data transfer [34], it has not yet been fully explored. In this work, we address a broader question: what does the support tell us about the detection? We show that additional properties of an object can be retrieved from the support without changing the training or the detection procedure. To this end, we augment the codebook by some additional information to establish a link between the detection and the parts of different training examples. In particular, we demonstrate two important properties of the support:

Firstly, the different views of an object can be handled by a single codebook and even a single view-independent voting space since the support and its backprojection encode enough information to deal with the viewpoint variations. This is very relevant in practice since state-of-the-art GHT-based multi-view detectors like [56] treat different viewpoints as different classes and train a battery of one-vs-the-rest codebooks for each view. The training and detection time of these approaches scale thus linearly with the number of quantized views. Unlike these approaches, we train a single shared codebook for all views, i.e. the complexity of training and matching a feature against it is independent of the number of quantized views. The proposed training procedure also allows us to make better use of training data by sharing features of different views. Having the shared codebook and depending on the availability of view annotations and amount of training data, two voting schemes are proposed which outperform the battery of one-vs-the rest detectors both in terms of accuracy and computational complexity, in particular, when only few training examples are available.

Secondly, not only we can detect objects under large view variations with a single shared codebook, but also we can use the support for defining similarity measures to retrieve nearest examples. One can then estimate continuous parameters (like object’s pose or view) of detections by the parameters of the nearest examples. To this end, we introduce two similarity measures based on the support. The first one efficiently measures the similarity of a detection to all training examples. We show that this measure is applicable to retrieve various continuous parameters like the pose of a detected pedestrian. The second metric finds dense feature correspondences and can be used as a similarity measure between any two detections. This measure is particularly interesting as it is part-based and

naturally takes the spatial relationships of the parts into account. It also inherits nice properties like good generalization and insensitivity to partial occlusions from the ISM model. The power of the latter similarity metric is demonstrated in the task of view-retrieval in presence of partial occlusions.

## 2 Related Work

Several approaches have been proposed for the creation of codebooks and to learn the mapping of image features into the Hough space. While [3] clusters the sparse image features only based on appearance, the spatial distribution of the image features is also used as a cue for the clustering in [7,8]. [9] proposes to store training features without clustering and use them as a codebook. Hough forests [10] use a random forest framework instead of clustering for codebook creation.

In order to vote with a codebook, matched codewords cast weighted votes. While the work of [3] uses a non-parametric Parzen estimate for the spatial distribution and thus for the determination of the weights, [11] re-weights the votes using a max-margin framework for better detection. The voting structure has been addressed in [12], where voting lines are proposed to better cope with scale-location ambiguities.

In [3], the backprojection has been used for verification. To this end, the training data is segmented and the local foreground-background masks are stored with the codebook. When a maximum is detected in the voting space, the local segmentation masks are used to infer a global segmentation for the detection in the image. The global segmentation is then in turn used to improve recognition by discarding votes from background and reweighting the hypothesis. In [4], the individual parts of an object (e.g. front wheel of a motorbike) are also annotated in the training and used to infer part-labels for a test instance. In this work, we neither focus on hypothesis verification nor require time consuming segmentations and part annotations of the training data.

The handling of multiple object views has also been addressed in the literature, see e.g. [13]. Based on the ISM and a silhouette-based verification step [3], the model can be extended by handling the viewpoint as additional dimension. Thomas et al. [5] train a codebook for each annotated view and link the views together by appearance. [14] extends the voting scheme of ISM by a 4th dimension, namely shape represented by silhouettes, to improve the segmentation based verification step. Although this approach uses a shared codebook for multi-aspect detection of pedestrians, it is limited as it only considers pedestrians which are already handled by the ISM. Other approaches use the statistical or geometric relationships between views in the training data to reason about the 3D structure of the object [15,16,17,18]. Since these approaches need many viewpoints of an object, they are very expensive in data. Although some of the missing data can be synthesized from existing data by interpolation [16,19], the interpolation still requires a certain number of views in the training data. In [20], a discriminative approach is proposed to handle aspects more general than for a specific class.

To this end, latent variables that model the aspect are inferred and the latent discriminative aspect parameters are then used for detection.

### 3 Multi-view Localization with ISMs

The object detection in our approach is based on the Implicit Shape Model (ISM) [3]. In this framework, training consists of clustering a set of training patches  $P^{train}$  to form a codebook of visual appearance  $\mathcal{C}$  and storing patch occurrences for each codebook entry. At runtime, for each test image  $\mathcal{I}_{test}$ , all test patches  $P^{test}$  are extracted and matched against  $\mathcal{C}$ . Each patch casts weighted votes for the possible location of the object center. These votes are based on the spatial distribution of the matching codebook entry to a Hough accumulator  $\mathcal{V}$ . This encodes the parameters of the object center, e.g. position and scale in the image domain; see Fig. 1(a). This way all the votes are cast in  $\mathcal{V}$  (Fig. 1(b)). The probability of the presence of the object center at every location of  $\mathcal{V}$  is estimated by a Parzen-window density estimator with a Gaussian kernel. Consistent configurations are searched as local maxima in  $\mathcal{V}$  forming a set of candidates  $\mathbf{v}_h \in \mathcal{V}$ . For each candidate  $\mathbf{v}_h$ , its support  $S_h$  is formed by collecting all the votes contributing to its detection.

In the following, we discuss the training of the shared codebook for all the views in detail and explain the necessary augmentation of the codebook entries. Then multi-view detection with this codebook is discussed. Afterwards, we look into the support,  $S_h$ , and its backprojection to the image domain for bounding box estimation and retrieving nearest training examples. Finally, we introduce a similarity metric based on the support for comparing two arbitrary detections.

#### 3.1 Training a Shared Codebook

To train the codebook  $\mathcal{C}$  with entries  $c_1 \dots c_{|\mathcal{C}|}$ , a set of training patches are collected  $P^{train}$ . Training patches are sampled from a set of bounding box annotated positive images and a set of background images. Each training patch,  $P_k^{train} = (\mathcal{I}_k, l_k, \mathbf{d}_k, \theta_k)$ , has an appearance  $\mathcal{I}_k$ , class label  $l_k$ , a relative position to the object center and an occurrence scale  $\mathbf{d}_k = (x_d, y_d, s_d)$ , and additional training data information  $\theta_k$ , e.g. the identity of the training image it is sampled from.

For building the codebook, we use the recently developed method of Hough Forests [10] as it allows us to use dense features and also due to its superior performance compared to other methods, e.g. the average-link clustering used in [3]. Hough Forests are random forests [21] which are trained discriminatively to boost the voting performance. During training, a binary test is assigned recursively to each node of the trees that splits the training patches into two sets. Splitting is continued until the maximum depth is reached or the number of remaining patches in a node is lower than a predefined threshold (both fixed to 20 in the current implementation). The codebook consists of the leaves which store the arrived training patches  $P_k^{train}$ .

The binary tests are selected using the same two optimization functionals as [10], to reduce the uncertainty of class-labels and offset. The view annotations are completely discarded. In our implementation, the label of each patch,  $l_k$ , is a binary value assigned to one if the patch was drawn from inside the bounding box of a positive image and otherwise it is set to zero. However, all training data information, including the label, can be recovered from  $\theta_k$ .

### 3.2 Multi-view Detection

For detection of objects in a test instance, every patch of the test instance  $P_i^{test}$  is matched against the codebook  $\mathcal{C}$  and its probabilistic votes are cast to the voting space  $\mathcal{V}$ . In particular, by matching the patch  $P_i^{test} = (\mathcal{I}_i, x_i, y_i, s_i)$  to the codebook, the list of all occurrences  $\mathcal{O}_i = \{o = (\mathcal{I}, l, \mathbf{d}, \theta)\}$  stored in the matching entries is obtained. In this paper, we propose two schemes for casting these votes to  $\mathcal{V}$ : *joint voting* and *separate voting*.

**Joint Voting:** Votes are cast to a 3D voting space  $\mathcal{V}(x, y, s)$ . Let us denote the proportion of positive (same label) to negative (different label) patches of each codebook entry by  $r_c^{pos}$  and the number of positive occurrences by  $n_c^{pos}$ . Then for each occurrence  $o = (\mathcal{I}, l, \mathbf{d}, \theta) \in \mathcal{O}_i$ , a vote with weight  $w_o$  is cast to the position  $\mathbf{v} = (v_1, v_2, v_3)$ :

$$v_1 = x_i - x_d(s_i/s_d) \quad (1)$$

$$v_2 = y_i - y_d(s_i/s_d) \quad (2)$$

$$v_3 = s_i/s_d \quad (3)$$

$$w_o = r_c^{pos} / n_c^{pos} \quad (4)$$

After all votes are cast, the local maxima of  $\mathcal{V}$  are found, forming a set of candidate hypotheses. In this voting scheme, votes from different training examples and from different views can contribute to a hypothesis and detection is performed without using the viewpoint annotations.

**Separate Voting:** Voting in *separate voting* is performed in a 4D voting space  $\mathcal{V}(x, y, s, view)$ . For each view  $view$ , let us denote the number and the proportion of positive occurrences in  $c$  with label  $view$  by  $n_c^{view}$  and  $r_c^{view}$ , respectively. Then for each occurrence  $o$ , a vote is cast to the position  $\mathbf{v} = (v_1, v_2, v_3, v_4)$  with weight  $w_o$ :

$$v_1 = x_i - x_d(s_i/s_d) \quad (5)$$

$$v_2 = y_i - y_d(s_i/s_d) \quad (6)$$

$$v_3 = s_i/s_d \quad (7)$$

$$v_4 = view \quad (8)$$

$$w_o = r_c^{view} / n_c^{view} \quad (9)$$

The local maxima of  $\mathcal{V}$  are found after all votes are collected to form a set of candidate hypotheses. In this voting scheme, only votes from training examples of a particular viewpoint can contribute to a hypothesis of that view.

### 3.3 Detection Support and Backprojection

After the local maxima are localized in  $\mathcal{V}$ , the candidate hypotheses are determined in terms of their center position and scale (and view in separate voting). We then collect all votes in the support of a hypothesis (i.e. votes contributing to its detection) and exploit it to get additional information about it. For a hypothesis  $h$  in location  $\mathbf{v}_h \in \mathcal{V}$ , we define its support  $S_h$  as (see Fig. [1\(b\)](#)):

$$S_h = \{ \mathbf{v} \in \mathcal{V} | K(\mathbf{v} - \mathbf{v}_h) > 0 \} . \quad (10)$$

where  $K$  is a radially symmetric (in  $x$  and  $y$ ) kernel with only local support such that the set  $S_h$  contains only votes in the local neighborhood of  $\mathbf{v}_h$ . In the current implementation, only votes from the same scale, and same view in the case of separate voting, are considered as votes contributing in the support.

Additionally, we can define the backprojection as a mapping from  $\mathcal{V}$  to the image domain to form the backprojection mask  $\mathcal{M}$  (see Fig. [1\(c\)](#)):

$$B : \{ \mathbf{v} \in \mathcal{V} : \text{condition} \} \mapsto \mathcal{M} . \quad (11)$$

where *condition* are constraints on the votes. Having a constraint, e.g.  $\mathbf{v} \in S_h$ , the mask is constructed by projecting all the votes satisfying the constraint back to the image domain. Since each feature point  $\mathbf{x} = (x, y)$  in the test image is mapped to the voting space for detection, the mask can be calculated by mapping every vote  $\mathbf{v} = (v_1, v_2, v_3)$  back to  $\mathbf{x}$  with weight  $w_o$  (see Fig. [1\(d\)](#) for an example of such a mask). The total weight of a mask  $w_{\mathcal{M}}$  is then defined as the sum of the weights of all the votes mapped to it.

**Bounding box estimation from backprojection:** Most approaches (e.g. [3.6.10](#)) estimate the extent of the object’s presence by placing the average bounding box of training images scaled and translated to the detection center. Although this measure is sufficiently accurate for the rather generous standard evaluation criteria like [22](#), this measure is not applicable to multi-view detection with joint voting where aspect ratios of different views widely vary. Inspired by [3](#), we propose using the backprojection of the supporting features for this purpose. In our work, the backprojection mask is simply thresholded by an adaptive threshold (set to half the value range) to form a binary mask. The tightest bounding box encompassing this mask is used as our bounding box estimate. Of course this is an oversimplification and there is still the possibility of more sophisticated bounding box estimations, e.g. [23](#), but simple thresholding suffices to obtain reasonable bounding box estimates.

**Retrieving nearest training images:** By conditioning the back-projection of a hypothesis support  $S_h$  to the votes coming from a single training example with identity  $tr$ , one can measure how much  $tr$  contributes to the detection of  $h$ . Formally, we can write

$$B : \{ \mathbf{v} \in \mathcal{V} : \mathbf{v} \in S_h \wedge \theta_{\mathbf{v}} = tr \} \mapsto \mathcal{M}_{tr} . \quad (12)$$

The total weight of  $\mathcal{M}_{tr}$ ,  $w_{\mathcal{M}_{tr}}$ , can then be used as a holistic measure of similarity between the hypothesis  $h$  and the training image  $tr$ . In principle, by introducing additional constraints, one can enforce more specific similarity measures, e.g. similarity of the left side of  $h$  to the left side of  $tr$ . Since we sample only a sparse set of patches from the training examples during training, this measure establishes correspondences between sparse locations of the detection and a training example. Fig. 7 shows some examples of  $\mathcal{M}_{tr}$ .

**Support intersection as a metric:** Let  $I(o, S_h)$  be an indicator variable which is one if there is any vote in  $S_h$  which comes from occurrence  $o \in \mathcal{O}$  and zero otherwise. We define the support intersection of two hypotheses  $h_1$  and  $h_2$  as:

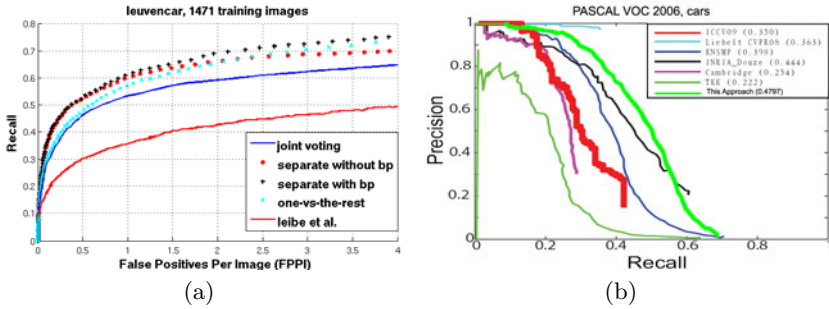
$$S_{h_1} \cap S_{h_2} = \frac{\sum_{o \in \mathcal{O}} w_o I(o, S_{h_1}) I(o, S_{h_2})}{\sum_{o \in \mathcal{O}} w_o I(o, S_{h_1})}. \quad (13)$$

Note that the similarity measure is not symmetric due to the normalization factor. Yet, this factor is important as it makes the measure independent of the detection weight and as it can also account for occluded regions. The support intersection can be used as a similarity measure between two detections. This similarity measure is a model-based similarity measure. There is a close link between the support intersection in (13) and the histogram intersection kernel used in bag-of-words image classification [24]. This said, there are also substantial differences between the two. Since the detection is done with ISM, the support of the detection takes the spatial relationships of the features into account. Therefore, there is no need for a fixed grid on top of the bag-of-words representation as in the spatial pyramid kernel [24]. In addition, this metric is part-based and benefits from the generalization capabilities of part-based methods and their insensitivity to occlusions, as shown in the experiments.

It is worthwhile to note an important difference between the similarity measures (12) and (13). The similarity measure in (12) can only be used to find the similarity between sparse patches of a detection and a training example, i.e. only matching to the same patches sampled during training. But support intersection establishes a dense feature correspondence between any two detections. Due to the dense correspondences in (13), for comparing two detections, this similarity measure has a computational cost in the order of the number of votes in its support. However, this is about the same cost it takes to consider sparse correspondences to all training examples in (12).

## 4 Experiments

In order to assess the performance of the multi-view detectors described in Sect. 3.2, we use three datasets. The multi-view Leuven-cars dataset [6] contains 1471 training cars annotated with seven different views and a sequence of 1175 images for testing. The multi-view Leuven-motorbikes dataset [5] contains 217 training images annotated with 16 quantized views and 179 test images. And the PASCAL VOC'06 cars dataset [22]. Further experiments for nearest neighbor



**Fig. 2.** (a) Detection performance for the Leuven-cars dataset and comparison to Leibe et al. [6]. Separate voting with bounding boxes estimated from backprojection (bp) achieves the best performance despite much lower training and detection complexity than baseline (one-vs-the-rest). Joint voting with even lower detection complexity and without using view annotations gives competitive results. (b) Performance comparison of joint-voting with state-of-the-art approaches on PASCAL VOC 2006 cars dataset.

retrieval are carried out on the TUD-pedestrians dataset introduced in [25] and the cars datasets. The TUD-pedestrians dataset provides 400 training images and 250 images for testing. Throughout the experiments, only bounding box annotations of the training images are used. The segmentation masks that are provided for some datasets are discarded.

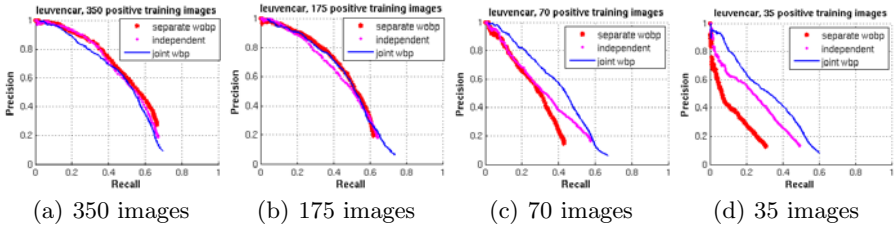
#### 4.1 Multi-view Detection

As a baseline comparison for the multi-view detection, we consider the popular one-vs-the-rest detector. For each view, the training is carried out with the positive training images of a view versus random patches from the Caltech 256 clutter set plus all the positive training images of the other views. An edge detector has been carried out both for training and testing and only features with their center on an edge are considered.

In order to make fair comparisons, the training and detection parameters are kept the same throughout the experiments. In particular, the number of trees in each forest is set to 15. From each training image 100 patches are sampled and the number of background patches is kept constant at 20000 patches. For detection, the kernel used for the density estimation is a Gaussian with  $\sigma = 2.5$  and the first 20 local maxima per image are considered. When the backprojection is not used for bounding box estimation, non-maxima suppression is done by removing all detections whose centers are within the bounding box (with 90% of its size) of another detection with a higher weight. When using backprojection, the hypothesis with the highest weight is included and its features are removed from all other hypotheses, thereby decreasing their weights.

The results of this experiment are shown in Fig. 2. As can be seen, separate voting with the help of backprojection performs best and estimating the bounding box with backprojection slightly increases the performance of the system.

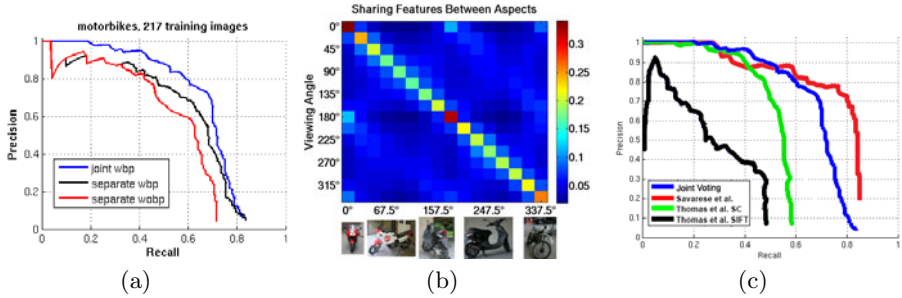




**Fig. 3.** The effect of training size on the performance of joint voting, separate voting, and a battery of independent one-vs-the-background classifiers: With the abundance of training data per view, the separate-voting works best. The advantage of sharing is significant with lower number of training examples, especially compared to the separate voting with an identical codebook although no view annotations are used. Joint and separate voting outperform the independent detector in efficiency and/or accuracy.

Joint voting also shows a competitive performance. It is worthwhile to note that the superior performance of separate voting is mainly due to the abundance of training images per view in this dataset and the presence of additional view information not used by joint voting. By sharing features across views, as e.g. shown in the work of Torralba et al. [26], one expects to benefit mainly when the training data is limited. In order to verify this, we did the following experiment.

We compare the performance of joint voting, separate voting, and a battery of independent one-vs-the-background classifiers for 50, 25, 10, and 5 training images per view (7 views). In all the experiments, the full background set from Caltech 256 clutter is used and the set of training images for all three detectors is identical. Joint voting and separate voting use identical shared codebooks whereas a separate codebook is trained per view for the independent detector (see Fig. 3). With fewer training examples, as expected, the performance of all three detectors degrades, but that of joint voting far more gently. In particular, the comparison of separate voting and joint voting for few training images is very interesting. Although an identical codebook is used, joint voting significantly outperforms separate voting. This performance gap seems to narrow by using several codebooks (number of views) and thus more codebook entries for the independent detector but the performance of joint voting is still superior in terms of accuracy as well as training and detection time. In order to assess performances on a more challenging dataset, we evaluated joint voting and separate voting for the Leuven-motorbikes dataset [5] where the test set is provided by the PASCAL VOC Challenge [27]. The motorbikes have more variability in their appearance and the views are quantified finer because of the larger variability in aspect ratios. For the sake of a fair comparison the same training and test settings as in [5] is used. The results of this experiment are shown in Fig. 4(a). Note that the detection result with joint voting is obtained only using the bounding box annotations for the training data and using no view annotations. It is important to note that the aim of the experiments is to show improvements over our baselines with the same parameters throughout experiments. The performance of joint voting and other state-of-the-art approaches is shown in Fig. 4(c) to give



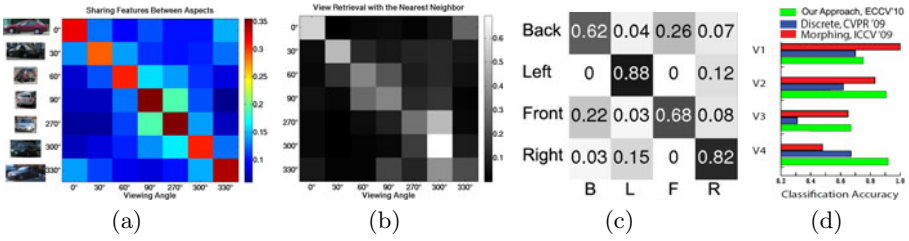
**Fig. 4.** (a) Joint voting achieves better results than separate voting because of insufficient training data per view and finely quantized views although it does not use view annotations. Estimating the bounding box from backprojection even leads to a small improvement. (b) Sharing of features across views of motorbikes. (c) Performance comparison to state-of-the-art multi-view detectors for the motorbikes dataset.

the reader an idea of the performance of other approaches compared to ours on this dataset. Note that in Thomas et al. [5] and [17] pixel accurate segmented training data is used. In contrast to our approach and [5], the sliding window approach of Savarese et al. [17] explicitly uses the geometrical relationships of different views. Although these relationships seem to be useful (better recall) for detection it comes at high computational costs which makes this approach not scalable to large datasets. In particular, testing with this approach has linear complexity in the number of training examples compared to logarithmic in our implementation. And training complexity is quadratic in the number of training images (linear in our case). In addition, although this work does not use view annotations, unlike our approach it needs many views of several training examples which are expensive to acquire.

**Sharing features across views.** One of the main advantages of training multiple views jointly is the sharing of features. In order to evaluate the capability of our method in doing so, we are creating a sharing matrix of size  $n_{views} \times n_{views}$ . Each element of this matrix shows, on average, how many features of the column view are used for detection of the row view. Since the test set of none of the datasets is annotated for views, this experiment is done on the set of training data with a leave-one-out strategy. When running the detector on a training instance, we are removing all the occurrences that are originating from that instance from the forest. The sharing matrices for the Leuven-cars and Leuven-motorbikes datasets are shown in Figs. 4(b) and 5(a).

## 4.2 Estimating View-Point with Nearest Neighbors

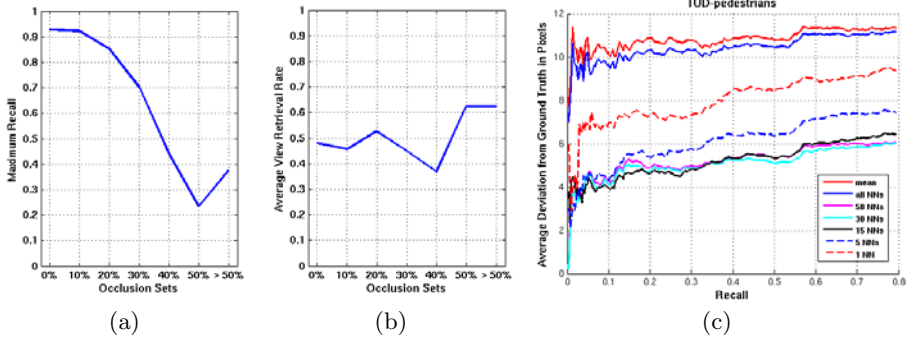
As described in Sect. 3.3, support intersection can be used as metric to compare two detections. In order to assess the quality of this metric, we use it to retrieve the viewpoint of the detected cars in the Leuven and PASCAL VOC'06 cars datasets. To this end, we have hand-annotated the viewpoint of the full



**Fig. 5.** (a) Sharing codebook occurrences across views for Leuven-cars. (b) Viewpoint retrieval with the nearest neighbor using (13). (c) View-point classification for detected cars in the VOC'06 dataset. As can be seen, the confusion appears to be limited only to similar views. (d) Comparison to state-of-the-art (16,18) for view classification on VOC'06 (for a fair comparison detections up to 43% recall are considered; average accuracy 82%). Note that our nearest neighboring approach leads to superior performance and more balanced estimation. Comparing the sharing pattern and view confusions is also interesting; e.g. front and back views share many features but their view have been separated well. This shows the presence of additional information in the support.

Leuven-cars test set. For the PASCAL VOC'06 cars set, the ground truth annotations were used. For the Leuven-cars, we have run the detector on the positive set of the training data and collected a set of detections. For the VOC'06 set, the same procedure is carried out but on the validation set. All detections are done with joint voting (see Sect. 3.2) and not using view annotations. By comparing the support of a test detection to the support of all positive collected detections using (13), the nearest neighbor is retrieved and the estimated view of it is assigned to the test detection. This has been done for all the true positives in the test set and their estimated viewpoint is stored. By comparing the estimated viewpoint with the ground truth annotations, the confusion matrix in Fig. 5(b) (with average diagonal of 43%) is created where the rows indicate the ground-truth viewpoints and columns are the estimated viewpoints. In order to see if retrieving more nearest neighbors would add robustness to this process, this experiment is repeated by retrieving the 35 nearest training detections for each test detection and assigning the viewpoint of the majority to it (with average diagonal of 50%). The results for the VOC'06 are given in Fig. 5(c,d). As can be seen, most confusion is happening between very similar views. Note that the features used in our detection system are relatively invariant with respect to small viewpoint changes and the training is done without using viewpoint annotations and in a way to optimize detection performance. In addition, there is a relatively large overlap in the annotation of nearby views due to the difficulty of precise viewpoint estimation even for humans. A video showing the estimated views for the entire Leuven-cars dataset is available under <http://www.vision.ee.ethz.ch/~nrazavi>.

**The effect of occlusion:** In order to assess the quality of the support intersection similarity metric in the presence of occlusions, we have annotated all the cars in every tenth frame of the Leuven-cars sequence based on the amount of



**Fig. 6.** (a-b) The view retrieval performance using (13) together with the proportion of the cars detected depending on the amount of occluded regions for a subset of the Leuven-car sequence. (the last two sets, 50% and > 50%, have very few instances). The recall and view retrieval performances were calculated independently for each occlusion set. Interestingly, although the detection performance deteriorates from large occlusions (a), the viewpoint retrieval performance is affected very little (b) which shows robustness of this similarity measure to occlusions. (c) Distance between the ankles estimated by the median of the  $k$  nearest training images using (12) compared to mean and median as baselines. The estimation is robust even at high recall rates.

occlusion: not occluded, 10%, 20%, 30%, 40%, and > 50% occluded regions. In this experiment, first the detector, with the same settings as in the multi-view experiment, Sect. 4.1, is applied to all the images and a number of detections are retrieved for each image. Then for each correct detection, its viewpoint is estimated as described above. For each occlusion set, we have evaluated how accurately the viewpoint is estimated. The results in Fig. 6(b) show the robustness of this nearest neighbor metric with respect to partial occlusions.

### 4.3 Retrieving Nearest Training Examples

In Sect. 3.3, we have explained how backprojection can be used as a similarity measure between object hypothesis and the training examples. In the following experiment, we are using such information to estimate the distance between the ankles of pedestrians as an indicator of their pose; see Fig. 7. We carried out our experiments on the TUD-pedestrians dataset. Training data of this dataset has annotations of the joint positions and this information is exploited for estimating the Euclidean distance (in pixels) between the ankles of a test instance. For the sake of evaluation, we have produced the same annotations for the test set. The distance between the ankles of the test instance is then estimated as the median of this distance in the  $k$  NNs. Figure 6(c) shows the deviation of the estimated distance from the ground truth for different values of  $k$ . As a baseline, we also show the deviation from the ground truth if the distance is estimated by the mean or median distance of the whole training set.



**Fig. 7.** Two test detections from TUD-pedestrians dataset and their top ten nearest training examples (top row; nearest examples ordered from left to right) and backprojections of detection support to them (bottom row) using (12). The blue box shows the estimated bounding box from the backprojection mask (blended). Note the similarity of the poses between the test instances and retrieved nearest training images.

## 5 Conclusions

We have introduced an extension of the Hough-based object detection to handle multiple viewpoints. It builds a shared codebook by considering different viewpoints jointly. Sharing features across views allows for a better use of training data and increases the efficiency of training and detection. The performance improvement of sharing is more substantial with few training data. Moreover, we have shown that the support of a detection and its backprojection can be exploited to estimate the extent of a detection, retrieve nearest training examples, and establish an occlusion-insensitive similarity measure between two detections.

Although the verification of object hypotheses is not the focus of this work, the detection performance is likely to improve by an additional verification step like MDL [3]. Moreover, the backprojection masks could be used in combination with a CRF to obtain object segmentations similar to [28]. The similarity metrics could be used in the context of SVM-KNN [29] for verification.

**Acknowledgments.** We wish to thank the Swiss National Fund (SNF) for support through the CASTOR project (200021-118106).

## References

1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *TPAMI* 26, 1475–1490 (2004)
2. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition* 13, 111–122 (1981)
3. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. *IJCV* 77, 259–289 (2008)
4. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Van Gool, L.: Using multi-view recognition and meta-data annotation to guide a robot’s attention. *Int. J. Rob. Res.* 28, 976–998 (2009)
5. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Gool, L.V.: Towards multi-view object class detection. In: *CVPR* (2006)
6. Leibe, B., Cornelis, N., Cornelis, K., Gool, L.V.: Dynamic 3d scene analysis from a moving vehicle. In: *CVPR* (2007)
7. Opelt, A., Pinz, A., Zisserman, A.: Learning an alphabet of shape and appearance for multi-class object detection. *IJCV* (2008)

8. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. *TPAMI* 30, 1270–1281 (2008)
9. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: *CVPR* (2008)
10. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: *CVPR* (2009)
11. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: *CVPR* (2009)
12. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: *ICCV* (2009)
13. Selinger, A., Nelson, R.C.: Appearance-based object recognition using multiple views. In: *CVPR* (2001)
14. Seemann, E., Leibe, B., Schiele, B.: Multi-aspect detection of articulated objects. In: *CVPR* (2006)
15. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3d object recognition. In: *CVPR* (2007)
16. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: *ICCV* (2009)
17. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: *ICCV* (2007)
18. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3d object classes. In: *CVPR* (2009)
19. Chiu, H.P., Kaelbling, L., Lozano-Perez, T.: Virtual training for multi-view object class recognition. In: *CVPR* (2007)
20. Farhadi, A., Tabrizi, M., Endres, I., Forsyth, D.: A latent model of discriminative aspect. In: *ICCV* (2009)
21. Breiman, L.: Random forests. *Machine Learning* 45, 5–32 (2001)
22. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The 2006 pascal visual object classes challenge (2006)
23. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 2–15. Springer, Heidelberg (2008)
24. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *CVPR* (2006)
25. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: *CVPR* (2008)
26. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. *TPAMI* 29, 854–869 (2007)
27. Everingham, M., et al.: The 2005 pascal visual object classes challenge (2005)
28. Winn, J.M., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: *CVPR*, vol. (1), pp. 37–44 (2006)
29. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: *CVPR* (2006)

# Multiple Instance Metric Learning from Automatically Labeled Bags of Faces

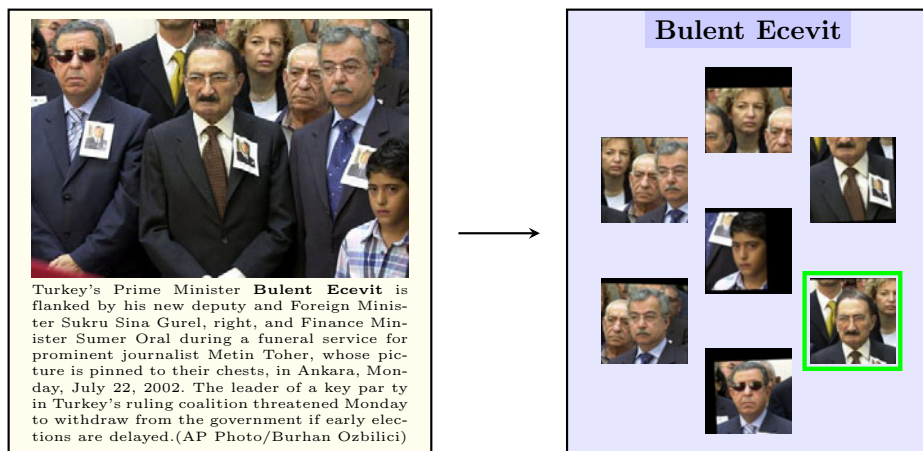
Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid

LEAR, INRIA, Grenoble  
Laboratoire Jean Kuntzmann  
firstname.lastname@inria.fr

**Abstract.** Metric learning aims at finding a distance that approximates a task-specific notion of semantic similarity. Typically, a Mahalanobis distance is learned from pairs of data labeled as being semantically similar or not. In this paper, we learn such metrics in a weakly supervised setting where “bags” of instances are labeled with “bags” of labels. We formulate the problem as a multiple instance learning (MIL) problem over pairs of bags. If two bags share at least one label, we label the pair positive, and negative otherwise. We propose to learn a metric using those labeled pairs of bags, leading to MildML, for multiple instance logistic discriminant metric learning. MildML iterates between updates of the metric and selection of putative positive pairs of examples from positive pairs of bags. To evaluate our approach, we introduce a large and challenging data set, *Labeled Yahoo! News*, which we have manually annotated and contains 31147 detected faces of 5873 different people in 20071 images. We group the faces detected in an image into a bag, and group the names detected in the caption into a corresponding set of labels. When the labels come from manual annotation, we find that MildML using the bag-level annotation performs as well as fully supervised metric learning using instance-level annotation. We also consider performance in the case of automatically extracted labels for the bags, where some of the bag labels do not correspond to any example in the bag. In this case MildML works substantially better than relying on noisy instance-level annotations derived from the bag-level annotation by resolving face-name associations in images with their captions.

## 1 Introduction

Metric learning is a supervised technique that finds a metric over a feature space that corresponds to a semantic distance defined by an annotator, who provides pairs of examples labeled with their semantic distance (typically zero or one). This semantic distance, in computer vision, might for instance express that two images depict the same object, or that they possess roughly the same layout. Once learned, the metric can be used in many different settings, *e.g.* for  $k$  nearest neighbor classification [1], matching and clustering samples based on the semantic similarity [2,3], or indexing for information retrieval and data visualization [4,5].



**Fig. 1.** Viewing news images with captions as a Multiple Instance Learning problem. The label “Bulent Ecevit” is assumed to be valid for at least one face in the face bag. The correct face image for Bulent Ecevit is highlighted in green.

Metric learning has recently received a lot of attention [1,3,6,7,8,9,10]. Most methods learn a Mahalanobis metric, which generalizes the Euclidean distance, using a variety of objective functions to optimize the metric. On the one hand, relatively large numbers of labeled pairs of examples are needed to learn Mahalanobis metrics, since the number of parameters scales quadratically with the data dimensionality. On the other hand, increasing the number of labeled pairs will immediately increase the run-time of metric learning algorithms, making large scale applications difficult. Regularization towards the Euclidean metric is often imposed to find a trade-off solution.

Large scale applications regularly arise in the field of computer vision, due to the explosive growth over the last decades of available data resulting from the advent of digital photography, photo sharing websites like Flickr and Facebook, or news media publishing online. Rarely, though, does this data come with clean annotations for the visual content. In an increasing number of cases, additional information relating to the images is nevertheless present, *e.g.* tags in Flickr or Facebook, captions for news images or surrounding text in web pages. Given this observation, a question that naturally arises is whether this massive quantity of weakly annotated data can be used to perform metric learning. Although weak forms of supervision have been considered for a number of computer vision related tasks [11,12], there is currently more work on metric learning from semi-supervised settings [2,13] than from noisy and weak supervision [14].

In this paper, we focus on a particular form of weak supervision where data points are clustered in small groups that we call bags. Bags appear naturally in several computer vision settings: for instance, an image can be viewed as a bag of several regions or segments [15] – each of which is described by a feature vector – or a video sequence as a bag of frames [14]. Multiple instance learning



(MIL) [16] refers precisely to the class of problems where data instances appear in bags, and each bag contains at least one instance for each label associated with the bag.

The work closest related to our is [17], where the authors learn a metric from MIL data for image auto-annotation. Compared to their settings, though, we will investigate the performance of metric learning when bag labels are noisy, which means that the underlying assumption of MIL will not be true in general: a bag may be assigned a label for which none of the instances is relevant.

More precisely, we focus on the problem of metric learning for face recognition. The goal is to obtain a metric that relates to the identity of a person, despite wide variations in pose, expression, illumination, hair style, etc. In our setting, bags are images and instances are faces appearing in these images, as illustrated in Figure 1. The labels are names automatically extracted from the image caption. As we will see, in this setting the handling of noisy labels can be viewed as a constrained clustering problem. Constrained clustering of faces using noisy name-labels has been considered by various authors [18,19,20,21,22], but these approaches do not integrate this with metric learning, except [2].

The paper is organized as follows: first we describe LDML [3], a recent state-of-the-art metric learning algorithm. Next, we propose a MIL formulation for learning similar types of metrics but on bag-level annotations which are potentially noisy. We refer to it as MildML, for Multiple Instance Logistic Discriminant Metric Learning. Then, we show how LDML can be adapted to infer instance-level annotations from bag-level annotation, and how this can help handle noisy data. In Section 3, we present the *Yahoo! News* data set, and our annotation of it that is publicly available as the *Labeled Yahoo! News* data set, and our feature extraction procedure. Section 4 presents our experimental results, and we conclude in Section 5.

## 2 Metric Learning from Various Levels of Annotation

In this section we show how Mahalanobis metrics can be learned from strong instance-level annotations to weaker bag-level forms of supervision. A Mahalanobis distance  $d_{\mathbf{M}}$  on  $\mathbb{R}^D$  generalizes the Euclidean distance, and for  $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$  it is defined as

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j), \quad (1)$$

where  $\mathbf{M}$  is a  $D \times D$  symmetric positive semidefinite matrix, *i.e.*  $\mathbf{M} \in \mathcal{S}_D^+$ . Note that  $\mathcal{S}_D^+$  is a cone in  $\mathbb{R}^{D \times D}$ , and therefore is a convex subset of  $\mathbb{R}^{D \times D}$ .

Below, we first describe LDML, a recent supervised metric learning method in Section 2.1, and modify it to learn low-rank matrices  $\mathbf{M}$ . In Section 2.2 we introduce MildML, a MIL extension of LDML that can handle noisy bag-level annotations. Then, in Section 2.3 we cast MIL metric learning as an joint metric learning and constrained clustering problem.

### 2.1 Supervised Metric Learning

In fully supervised settings, data points  $\mathbf{x}_i$  are manually associated with their true class labels coded by a binary vector  $y_i \in \{0, 1\}^C$ , where  $C$  is the number of classes. Let us denote  $\mathbf{X}$  the  $D \times N$  matrix whose columns are the data vectors  $\mathbf{x}_i$ , and  $\mathbf{Y} = [y_i] \in \mathbb{R}^{C \times N}$  the label matrix. Typically, exactly one component of  $y_i$  equals 1 and all the other equal 0.

This is the classical metric learning setup, and it has been extensively studied [13,10]. Here, we focus on Logistic Discriminant Metric Learning (LDML) [3], which maximizes the concave log-likelihood  $\mathcal{L}$  of a logistic discriminant model. Considering the convexity of  $\mathcal{S}_D^+$ , the optimization problem is convex, and can be solved for example using projected gradient descent [23]. The objective of LDML is:

$$\underset{\mathbf{M}, b}{\text{maximize}} \quad \mathcal{L} = \sum_{i,j} t_{ij} \log p_{ij} + (1 - t_{ij}) \log(1 - p_{ij}), \tag{2}$$

where  $t_{ij}$  denotes the equality of labels  $y_i$  and  $y_j$ , *i.e.*  $t_{ij} = y_i^\top y_j$ , and

$$p_{ij} = p(y_i = y_j | \mathbf{x}_i, \mathbf{x}_j, \mathbf{M}, b) = \sigma(b - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)), \tag{3}$$

where  $\sigma(z) = (1 + \exp(-z))^{-1}$  is the sigmoid function, and the bias  $b$  acts as a threshold on the distance value to decide the identification of a new data pair.

We now modify LDML to learn metrics  $\mathbf{M}$  of fixed low rank, which reduces the number of free parameters and thus avoids over-fitting. As constraints on the rank of  $\mathbf{M}$  are non-convex, we can no longer use methods for convex optimization. Instead, we choose to define  $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$ , where  $\mathbf{L}$  is a  $d \times D$  matrix, which ensures that  $\mathbf{M}$  is a positive semidefinite matrix of rank  $d$ . We now optimize  $\mathcal{L}$  w.r.t.  $\mathbf{L}$  using gradient descend, and resort to multiple random initializations to avoid poor local optima. The gradient of  $\mathcal{L}$  with respect to  $\mathbf{L}$  equals

$$\frac{\partial \mathcal{L}}{\partial \mathbf{L}} = \mathbf{L} \sum_{i,j} (t_{ij} - p_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{4}$$

$$= 2\mathbf{L} \sum_i \mathbf{x}_i \left( \left( \sum_j t_{ij} - p_{ij} \right) \mathbf{x}_i^\top - \sum_j (t_{ij} - p_{ij}) \mathbf{x}_j^\top \right) \tag{5}$$

$$= 2\mathbf{L}\mathbf{X}\mathbf{H}\mathbf{X}^\top, \tag{6}$$

where  $\mathbf{H} = [h_{ij}] \in \mathbb{R}^{N \times N}$  with  $h_{ii} = \sum_{j \neq i} (t_{ij} - p_{ij})$  and  $h_{ij} = p_{ij} - t_{ij}$  for  $j \neq i$ . The gradient as in Equation 6 can be computed in complexity  $O(N(N+D)d)$  which, since  $d \ll D$ , is significantly better than the original LDML projected gradient, whose complexity is  $O(N(N+D)D+D^3)$ .

Note that the rows of  $\mathbf{L}$  can be restricted to be in the span of the columns of  $\mathbf{X}^\top$ . This is possible since this is true for the gradient (Equation 6) and since the Mahalanobis distance over the training data is invariant to perturbations of  $\mathbf{L}$  in directions outside the span of  $\mathbf{X}$ . Hence, using  $\mathbf{L} = \mathbf{A}\mathbf{X}^\top$ , we can write

the Mahalanobis distance in terms of inner products between data points, which allows us to use kernel functions to perform non-linear LDML like was done in [9]. Straightforward algebra shows that to learn the coefficient matrix  $\mathbf{A}$  we simply replace  $\mathbf{X}$  with the kernel matrix  $\mathbf{K}$  in the learning algorithm, which will then output the optimized matrix  $\mathbf{A}$ . In Section 4 we only report results using linear LDML; preliminary results using polynomial kernels did not show improvements over linear LDML.

## 2.2 Multiple Instance Metric Learning

We now consider the case where the individual labels  $y_i$  are unknown. Instead of supervision on the level of single examples, or pairs of examples, we assume here that the supervision is provided at the level of pairs of bags of examples. This naturally leads to a multiple instance learning formulation of the metric learning problem, which we refer to as MildML, for Multiple Instance Logistic Discriminant Metric Learning.

Let us denote a bag of examples as  $\mathcal{X}_d = \{\mathbf{x}_1^d, \mathbf{x}_2^d, \dots, \mathbf{x}_{N_d}^d\}$ , where  $N_d$  is the number of examples in the bag. The supervision is given by labels  $t_{de} \in \{0, 1\}$  that indicate whether for a pair of bags  $\mathcal{X}_d$  and  $\mathcal{X}_e$  there is at least one pair of examples  $(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X}_d \times \mathcal{X}_e$  such that  $\mathbf{x}_1$  and  $\mathbf{x}_2$  belong to the same class. If there is such a pair of examples then  $t_{de} = 1$ , and  $t_{de} = 0$  otherwise.

The objective in Equation 2 is readily adapted to the MIL setting by extending the definition of the distance to compare bags [17] with:

$$d_{\mathbf{M}}(\mathcal{X}_d, \mathcal{X}_e) = \min_{\mathbf{x}_1 \in \mathcal{X}_d, \mathbf{x}_2 \in \mathcal{X}_e} d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2). \quad (7)$$

which, using  $p_{de} = \sigma(b - d_{\mathbf{M}}(\mathcal{X}_d, \mathcal{X}_e))$ , leads to the following optimization:

$$\underset{\mathbf{M}, b}{\text{maximize}} \quad \mathcal{L} = \sum_{d, e} t_{de} \log p_{de} + (1 - t_{de}) \log(1 - p_{de}). \quad (8)$$

This objective makes bags that share a label closer, and pushes bags that do not share any label apart. For a negative pair of bags, all the pairs of instances that can be made from these two bags are pushed apart since the pair of examples with minimum distance is.

We optimize the objective iteratively by alternating (i) the pair selection by the min operator for a fixed metric, and (ii) optimizing the metric for a fixed selection of pairs. The optimization in the second step is exactly of the same form as the optimization of the low-rank version of LDML presented in the previous section. At each iteration step, we perform only one line search in the direction of the negative gradient, such that the pair selection is performed at each step of the gradient descent. This way we do not spend many gradient steps optimizing the metric for a selection of pairs that might still change later.

Note that since MildML does not try to specifically assign labels to instances, and instead for each pair of bags only a single pair of instances is used to learn the metric. The benefit is that this single pair is robust to noise in the data, but the drawback is that many pairs of examples are lost, especially the negative ones occurring inside a bag, which may impact the quality of the learned metric.

### 2.3 Estimating Instance Labels from Bag-Level Labels

In this section we consider the setting where we have partial knowledge of the labels of the instances in a bag  $\mathcal{X}_d$ , given by a label vector  $y_d \in \{0, 1\}^C$ , where  $y_d^{(n)} = 1$  indicates that the bag contains at least one example of class  $n$ . This setting is also known as Multiple Instance Multiple Label Learning (MIML). MildML is directly applicable in this case by defining  $t_{de} = 1$  if  $y_d^\top y_e \geq 1$ , and  $t_{de} = 0$  otherwise. On the other hand, LDML must be adapted to explicitly estimate the labels of the instances in each bag from the bag-level annotation. By estimating the instance labels we obtain a larger set of training pairs suitable for LDML, which may improve over the metric learned by MildML despite the possibly noisy annotation.

To learn a metric in this setting, we optimize the objective in Equation 2 jointly over the metric parameterized by  $\mathbf{L}$  and over the label matrix  $\mathbf{Y}$  subject to the label constraints given by the bag-level labeling:

$$\underset{\mathbf{Y}, \mathbf{L}, b}{\text{maximize}} \quad \mathcal{L} = \sum_{i,j} (y_i^\top y_j) \log p_{ij} + (1 - y_i^\top y_j) \log(1 - p_{ij}). \tag{9}$$

Unfortunately, the joint optimization is intractable. For fixed  $\mathbf{Y}$ , it is precisely the optimization problem discussed in Section 2.1. When optimizing  $\mathbf{Y}$  for fixed  $\mathbf{L}$  and  $b$ , we can rewrite the objective function as follows:

$$\mathcal{L} = \sum_{i,j} (y_i^\top y_j) (\log p_{ij} - \log(1 - p_{ij})) + c = \sum_{i,j} w_{ij} (y_i^\top y_j) + c, \tag{10}$$

where  $c = \sum_{i,j} \log(1 - p_{ij})$  and  $w_{ij} = b - d_M(\mathbf{x}_i, \mathbf{x}_j)$  are constants. This optimization problem is NP-hard, and we therefore have to resort to approximate optimization techniques.

Observing that the only non-constant terms in Equation 10 are those for data points in the same class, we can rewrite the objective for a particular instantiation of  $\mathbf{Y}$  as

$$\underset{\mathbf{Y}}{\text{maximize}} \quad \sum_{n=1}^C \sum_{i \in \mathcal{Y}_n} \sum_{j \in \mathcal{Y}_n} w_{ij}, \tag{11}$$

where  $\mathcal{Y}_n$  is the set of indices of instances that are assigned to class  $n$ , *i.e.*  $\mathcal{Y}_n = \{i | y_i^{(n)} = 1\}$ . Equation 11 reveals that we are solving a constrained clustering problem: we have to assign the instances to clusters corresponding to the classes so as to maximize the sum of intra-cluster similarities  $w_{ij}$ . The non-zero entries in the bag-level labels  $y_d$  define the subset of possible clusters for the instances in a bag  $\mathcal{X}_d$ . If we have  $y_d^\top y_e = 0$  for a pair of bags, this implies cannot-link constraints between all instance pairs that can be constructed from these bags.

To obtain an approximate solution for  $\mathbf{Y}$  we perform a form of local optimization. The label optimization is initialized by assigning all instances in a bag to each permissible class according to the bag label. We then maximize  $\mathcal{L}$  w.r.t. the labels of the instances in each bag in turn, also enforcing that each instance is

assigned to exactly one class. The optimization at bag level can be done exactly and efficiently using bipartite graph matching [24].

In our experiments we compare the method presented in the current section and MildML. Since both optimize an objective function based on LDML, differences in performance will be due to the strategy to leverage the bag-level labels: either by selecting a single pair of instances for each pair of bag (MildML) or by inferring instance level labels.

The method presented in this section is similar to the one presented in [17]. That work also infers instance level labels to perform metric learning in a MIL setting. However, it is based on estimating prototypes, or cluster centers, for each of the classes. The objective then tries to ensure that for each bag and each class label of the bag, there is at least one instance of the bag close to one of the centers of the class. A second term in the objective function forces the centers of different classes to be maximally separated. The optimization scheme is relatively complex, as in each iteration it involves minimizing a non-convex cost function. Due to this complexity and the fact that we are mainly interested in comparing the different strategies to leverage the bag-level annotation, we do not include [17] in our experimental evaluations.

### 3 Dataset and Feature Extraction

The *Yahoo! News* database was first introduced by Berg *et al.* [19], and was gathered in 2002–2003. It consists of news images and their captions describing the event appearing in the image. We produced a complete ground-truth annotation of the *Yahoo! News* database, which extends the annotation provided by the *Labeled Faces in the Wild* [25]. Our annotation not only includes more face detections and names, but also indicates which faces were detected in the same image, and which names were detected in the caption of that image. We coin our data set *Labeled Yahoo! News* [2].

**Face and Name Detection.** We applied the Viola–Jones face detector face detector on the complete *Yahoo! News* database to collect a large number of faces. The variations in appearances with respect to pose, expression, and illumination are wide, as shown in Figure 2. We kept all the detections, including the incorrect ones. In order to collect labels for the faces detected in each image, we ran a named entity recognition system [26] over the image captions. We also used the set of names from the *Labeled Faces in the Wild* data set as a dictionary for finding names in captions. Most often, the putative face labels collected in this manner include the correct name for all faces in an image, although this is not always true. Detected names without corresponding face detections are more common in the data set.

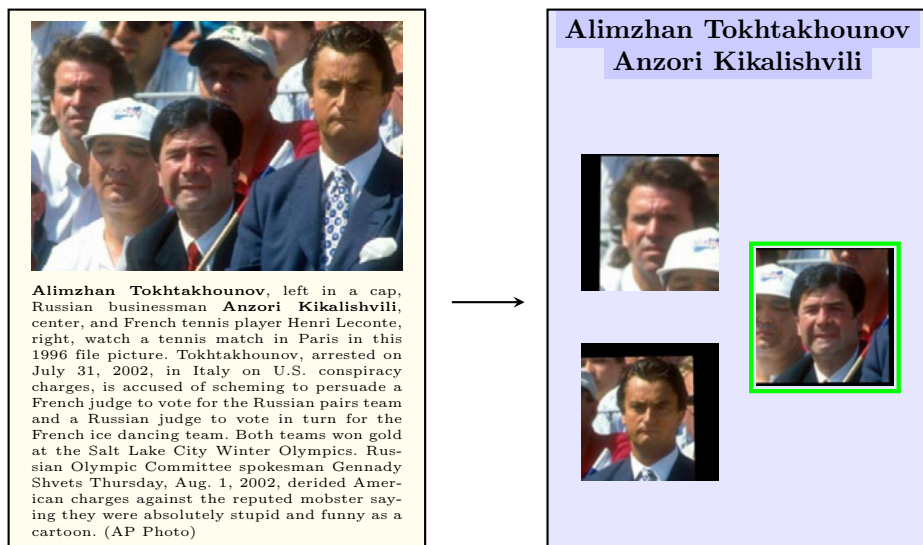
**Dataset Annotation.** Documents with no detected faces or names were removed, and we manually annotated the 28204 remaining documents for the correct

<sup>1</sup> Available online: <http://vis-www.cs.umass.edu/lfw/index.html>

<sup>2</sup> Available at <http://lear.inrialpes.fr/data/> together with the facial features.



**Fig. 2.** Examples of wide appearance variations in *Labeled Yahoo! News*



**Fig. 3.** On the left, example document from the *Labeled Yahoo! News* data set with the detected faces and labels on the right. Here, the automatic annotation is incorrect because the face of Alimzhan Tokhtakhounov was not detected. The correct face image for Anzori Kikalishvili is highlighted in green.

association between detected faces and detected names. For faces detections that are not matched to a name, the annotation indicates whether (a) it is a false detection (not a face), (b) it depicts a person whose name is not in the caption, or (c) it depicts a person whose name was missed by the name detector.

Likewise, for names that are not assigned to a face, the annotation indicates whether the face is present in the image but was missed by the detector. Finally, we also annotate the document when an undetected face matches an undetected name. Illustrations of resulting bags are given in Figure 1 and Figure 3.

**Definition of Test and Train Sets.** In order to divide this data set into completely independent training and test sets, we have proceeded the following way. Given the 23 person queries used in [24,27,28], the subset of documents containing these names is determined. This set is extended with documents containing “friends” of these 23 people, where friends are defined as people that

co-occur in at least one caption [28]. This forms set A. From the remaining set of documents we discard the 8133 ones that contain a name or a face from any person appearing in set A, such that it is now completely disjoint of set A.

Set A contains 9362 documents, 14827 faces and 1072 different names in the captions: because of the specific choice of queries, it has a strong bias towards politicians. Set B contains 10709 documents, 16320 faces and 4801 different people, relatively many athletes. The average number of face images for each person is significantly different between the two sets. Due to these differences between the two sets, we report performance by averaging the results obtained from training on either set and testing on the other.

**Facial Feature Extraction.** We computed a feature vector for each face detection in the following manner. First, we used the face alignment procedure of [29] to reduce effects due to differences in scale and orientation of the detected faces. Then, nine facial features are detected using the method of [20]. Around each of these nine points we extracted 128-d SIFT descriptors [30] on 3 different scales as in [3]. This results in 3456-d descriptors for every detected face.

## 4 Experimental Results

In this section we present our experimental results, and compare our different methods to learn metrics from weak supervision. The metrics are evaluated for two tasks: verification and clustering.

### 4.1 Metrics for Verification

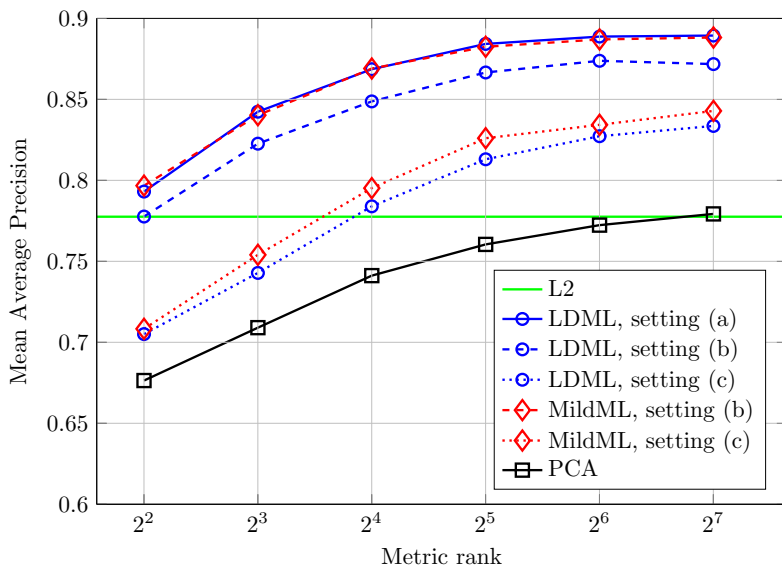
**Experimental Protocol.** In the face verification task we have to classify a pair of faces as representing the same person or not. Using our *Labeled Yahoo! News* data set, and following the evaluation protocol of [25], we sample 20000 face pairs of both sets A and B, approximately half of which are positives and half are negatives. Note that we can not use the same test set as *Labeled Faces in the Wild* because of overlap between test and train sets in this case. We measure average precision (AP) obtained at ten evenly spaced values of recall, and compute the mean AP (mAP) of (a) training the metric on set A and testing on set B’s pairs and (b) training the metric on set B to classify the pairs of set A.

**Experimental Results.** We study the performance of metric learning for different levels of supervision as described in Section 2. As baseline methods we consider the L2 metric in the original space and after applying PCA to reduce the dimensionality. We compare the following settings for learning metrics:

- (a) Instance-level manual annotations. This setting is only applicable to LDML, which should be an upper-bound on performance.
- (b) Bag-level manual annotations. This setting is applicable directly to MildML, and indirectly to LDML, using instance-level annotations obtained by applying constrained clustering using the L2 metric to define the face similarities.
- (c) Bag-level automatic annotations. Here, the labels are noisy, since the names in the caption do not necessarily correspond to faces detected in the images.

**Table 1.** Comparison of mean average precision on the *Labeled Yahoo! News* data set for LDML and LDML\* metrics. The two tables correspond to annotation settings (b) and (c), respectively. Please refer to the text for more details.

Setting (b)	Rank	4	8	16	32	64	128
LDML		77.8%	82.3%	84.9%	86.7%	87.4%	87.2%
LDML*		76.6%	82.4%	84.8%	86.5%	87.0%	87.0%
Setting (c)	Rank	4	8	16	32	64	128
LDML		70.5%	74.3%	78.4%	81.3%	82.7%	83.4%
LDML*		68.1%	73.0%	76.9%	79.2%	80.8%	81.3%



**Fig. 4.** Mean average precision for L2, PCA, LDML and MildML for the three settings (a), (b) and (c) described in the text, when varying the metric rank

In Figure 4, we report the performance of PCA, LDML and MildML for a wide range of dimensionalities and for the three settings (a), (b) and (c). As we increase the rank from  $d = 4$  to  $d = 128$ , we can see that the different methods reach a plateau of performance. For LDML with the instance-level annotations (a), the plateau is attained approximately at  $d = 32$ , with a performance of 88.4% of mAP, which is substantially above L2 and PCA metrics (77.9%).

When learning from manual bag-level annotations (b), we can still learn effective metrics: MildML and LDML are still substantially better than the L2 and PCA metrics. Moreover, MildML matches the performance of the fully supervised LDML on the entire range of metric ranks, with at most 0.6% of improvement for  $d = 4$  and 0.2% of decrease at  $d = 8$ . Notably, MildML outperforms the constrained clustering version of LDML using the same annotation (b), also over the range of metric ranks, by around 2 points.



When using the fully automatic annotation (c), performance drops for both methods, which is understandable since the labels are now noisy. For  $d \geq 16$ , the performance is still better than L2 and PCA. Also in this setting MildML performs best, reaching 84.3% for 128 dimensions. This score is closer to the fully supervised LDML (89.0%) than to the Euclidean distance (77.8%) or PCA (77.9%) for the same rank. Still, there is a significant gap between the supervised learning and the learning from automatically generated labels, and it appears that this gap narrows from low to higher dimensions: from 8.8% at  $d = 4$  to 4.5% at  $d = 128$  between the two levels of supervision for MildML.

Finally, we also considered a variant of LDML which re-estimates the instance level labels using the current metric, and iterates until convergence. We refer to this variant as LDML\*. As shown in Table 1, it has little influence on performance with the manual bag-level annotation of setting (b), at the cost of a much higher training time. On setting (c), the performance drops consistently by around 2%. We conclude that the noisy annotations penalize the clustering significantly. Remarkably, [17] also relies on data clustering while MildML does not.

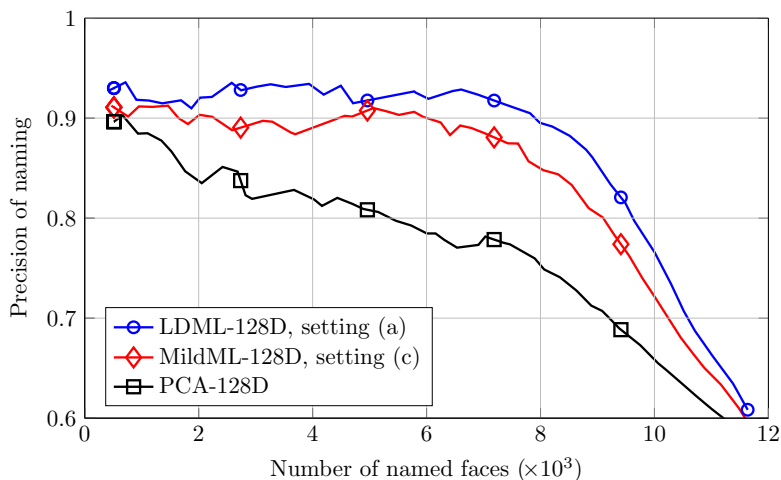
## 4.2 Metrics for Constrained Clustering

**Experimental Protocol.** In our second set of experiments, we assess the quality of the learned metrics for constrained clustering. We use the clustering algorithm described in Section 2.3 on one set of *Labeled Yahoo! News* after learning a metric on the other set. Note, the threshold  $b$  in Equation 1 directly influences the number of faces that are indeed associated to a label, *i.e.* named by our algorithm. Therefore, we can measure the precision (*i.e.* the ratio of correctly named faces over total number of named faces) of the clustering procedure for various numbers of named faces by varying the value of  $b$ . The curve is approximated on a reasonable range of named faces using a dichotomic search on the threshold value to obtain 50 approximatively evenly spaced points.

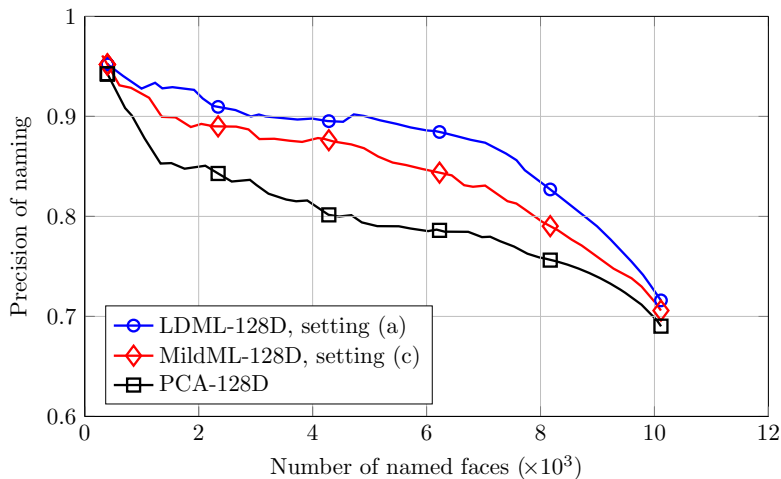
**Experimental Results.** We study the performance of metric learning for different levels of supervision as described in Section 2, while varying the parameter of the clustering algorithm. As a baseline method we consider PCA with 128 dimensions, which performs comparably to the L2 metric. In addition to PCA, we compare the following two learned metrics:

1. The fully supervised 128D LDML (which is comparable in performance to the 128D MildML learned from manual bag-level supervision).
2. The 128D MildML learned from automatically labeled bags of faces.

In Figure 5, we show the naming precision of those three metrics for the two sets: Figure 5(a) for clustering faces of set A, and (b) for set B. First, we notice that the clustering algorithm which associates each instance with a label is efficient, and is able to name several thousand faces with a precision above 80%. Second, there is a large increase of performance using learned metrics on both sets. LDML performs better than MildML, but the difference (of max. 6.0% between the two curves over the two sets) is smaller than the benefit of using MildML compared to PCA (up to +12.2%).



(a) Precision curve for clustering set A after learning metrics on set B.



(b) Precision curve for clustering set B after learning metrics on set A.

**Fig. 5.** Precision of the clustering algorithm on set A (top) and B (bottom) for three metrics of rank  $d = 128$  with the parameter varied, corresponding to a certain percentage of named faces. PCA is an unsupervised metric and performs worst. LDML is fully supervised at instance-level and performs best. MildML is learnt from automatically labeled bags and achieves performance close to the fully supervised metric.

## 5 Conclusion

In this paper, we have proposed a Multiple Instance Learning (MIL) formulation of metric learning to allow metric learning from data coming in the form of labeled bags. We refer to it as MildML, for multiple instance logistic discriminant metric learning. We have also shown that it is possible to extend LDML, a instance-level metric learning method, to learn from the same labeled bags using constrained clustering.

On the large and challenging *Labeled Yahoo! News* data set that we have manually annotated, we show that our proposed MildML approach leads to the best results when using bag-level labels. When the bag-level labels are noise-free, the results are comparable to the case where instance level labels are available. When using noisy bag labels, performance drops, but remains significantly better than that of the alternative methods. It appears that performing clustering to obtain instance-level labels and then learning LDML on the labeled examples does not perform well. The (costly) LDML\* procedure that iterates metric learning and instance label assignment does not remedy this problem.

In conclusion, we have shown that effective metrics can be learned from automatically generated bag-level labels, underlining the potential of weakly supervised methods. In future work we will consider learning algorithms that scale linearly with the number of data points, allowing learning from much larger data sets. Using larger data sets we expect the difference in performance between weakly supervised and fully supervised learning methods to diminish further.

## References

1. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2006)
2. Bilenko, M., Basu, S., Mooney, R.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML, p. 11. ACM, New York (2004)
3. Guillaumin, M., Verbeek, J., Schmid, C.: Is that you? Metric learning approaches for face identification. In: ICCV (2009)
4. Fu, Y., Li, Z., Huang, T., Katsaggelos, A.: Locally adaptive subspace and similarity metric learning for visual data clustering and retrieval. *Computer Vision and Image Understanding* 110, 390–402 (2008)
5. Jain, P., Kulis, B., Dhillon, I., Grauman, K.: Online metric learning and fast similarity search. In: NIPS (2008)
6. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS (2004)
7. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research* 6, 937–965 (2005)
8. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (2005)
9. Globerson, A., Roweis, S.: Metric learning by collapsing classes. In: NIPS (2006)
10. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)

11. Wang, J., Markert, K., Everingham, M.: Learning models for object recognition from natural language descriptions. In: BMVC (2009)
12. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
13. Wang, F., Chen, S., Zhang, C., Li, T.: Semi-supervised metric learning by maximizing constraint margin. In: Conference on Information and Knowledge Management (2008)
14. Yang, J., Yan, R., Hauptmann, A.: Multiple instance learning for labeling faces in broadcasting news video. In: ACM Multimedia (2005)
15. Zhou, Z., Zhang, M.: Multi-instance multi-label learning with application to scene classification. In: NIPS (2007)
16. Dietterich, T., Lathrop, R., Lozano-Perez, T., Pharmaceutical, A.: Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 31–71 (1997)
17. Jin, R., Wang, S., Zhou, Z.H.: Learning a distance metric from multi-instance multi-label data. In: CVPR (2009)
18. Satoh, S., Kanade, T.: Name-It: Association of face and name in video. In: CVPR (1997)
19. Berg, T., Berg, A., Edwards, J., Maire, M., White, R., Teh, Y., Learned-Miller, E., Forsyth, D.: Names and faces in the news. In: CVPR (2004)
20. Everingham, M., Sivic, J., Zisserman, A.: ‘Hello! My name is.. Buffy’ - Automatic naming of characters in TV video. In: BMVC (2006)
21. Holub, A., Moreels, P., Perona, P.: Unsupervised clustering for Google searches of celebrity images. In: IEEE Conference on Face and Gesture Recognition (2008)
22. Pham, P., Moens, M.F., Tuytelaars, T.: Linking names and faces: Seeing the problem in different ways. In: Proceedings of ECCV Workshop on Faces in Real-Life Images (2008)
23. Bertsekas, D.: On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control* 21, 174–184 (1976)
24. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Automatic face naming with caption-based supervision. In: CVPR (2008)
25. Huang, G., 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)
26. Deschacht, K., Moens, M.: Efficient hierarchical entity classification using conditional random fields. In: Proceedings of Workshop on Ontology Learning and Population (2006)
27. Ozkan, D., Duygulu, P.: A graph based approach for naming faces in news photos. In: CVPR, pp.1477–1482 (2006)
28. Mensink, T., Verbeek, J.: Improving people search using query expansions: How friends help to find people. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 86–99. Springer, Heidelberg (2008)
29. Huang, G., Jain, V., Learned-Miller, E.: Unsupervised joint alignment of complex images. In: ICCV (2007)
30. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* 60, 91–110 (2004)

# Partition Min-Hash for Partial Duplicate Image Discovery

David C. Lee<sup>1,\*</sup>, Qifa Ke<sup>2</sup>, and Michael Isard<sup>2</sup>

<sup>1</sup> Carnegie Mellon University

<sup>2</sup> Microsoft Research Silicon Valley

dclee@cs.cmu.edu, {qke, misard}@microsoft.com

**Abstract.** In this paper, we propose Partition min-Hash (PmH), a novel hashing scheme for discovering partial duplicate images from a large database. Unlike the standard min-Hash algorithm that assumes a bag of words image representation, our approach utilizes the fact that duplicate regions among images are often localized. By theoretical analysis, simulation, and empirical study, we show that PmH outperforms standard min-Hash in terms of precision and recall, while being orders of magnitude faster. When combined with the start-of-the-art Geometric min-Hash algorithm, our approach speeds up hashing by 10 times without losing precision or recall. When given a fixed time budget, our method achieves much higher recall than the state-of-the-art.

**Keywords:** partition min-hash, min-hash, partial duplicate image discovery.

## 1 Introduction

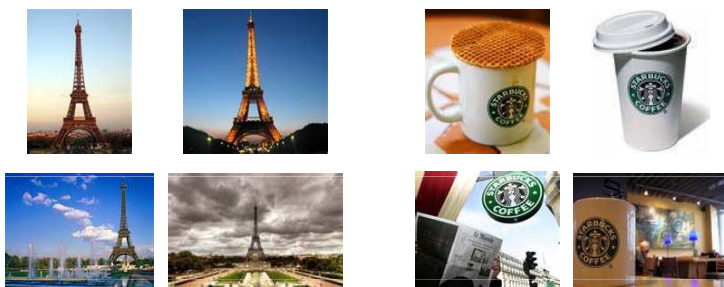
In this paper, we introduce a new method for partial duplicate image discovery in a large set of images. The goal of partial duplicate image discovery is to find groups of images in a large dataset that contain the same object, which may not necessarily occupy the entire image. Figure 1 shows examples of such groups of partial duplicate images. Partial duplicate image *discovery* differs from partial duplicate image *retrieval* in that there is no particular query image, but instead the task is to search for all groups of duplicate images in a dataset. Such a task is useful for identifying popular images on the web so that images can be ranked by their importance, for grouping similar images returned by a web search so that users can navigate the returned results more easily and intuitively, or for unsupervised discovery of objects.

Min-hash is a standard hashing scheme for discovering near-duplicate text documents or web pages [1]. Recently min-hash and its variants have been successfully applied to discovering near duplicate images [2,3], image clustering, image retrieval and object discovery [4]. In the min-hash algorithm, a hash function is applied to *all* visual words in an image ignoring the location of visual words, and the visual word with minimum hash value is selected as a global descriptor of the given image.

Unlike text documents which are usually represented by bag of words, images are strongly characterized by their 2D structure—objects are often spatially *localized* in the

---

\* This work was done during an internship at Microsoft Research Silicon Valley.



**Fig. 1.** Examples of partial duplicate images. The duplicate region occupies a small portion of each image.

image and there exist strong *geometric constraints* among the visual words in an object. Figure 1 shows some examples where the objects, and therefore the duplicate regions, are localized in the images. However, standard min-hash treats all visual words independently. A straightforward application of min-Hash to images ignores both locality and geometric constraints.

Geometric min-hash (GmH) [4] improves upon standard min-hash by considering the dependency among visual words. It first computes a min-hash value in a way similar to standard min-hash. The rest of the hash values in a sketch are then chosen only within a certain proximity of the first visual word. However, the locality property is still ignored in Geometric min-hash (GmH) when computing the first min-hash. If the first min-hash does not repeat between two matched images, the containing sketch is not repeatable and becomes useless.

Our ultimate goal is to detect partial-duplicate images from a web scale image database, where both precision/recall and computational cost are critical for scalability. In this paper, we aim to exploit locality and geometric constraints to improve precision/recall and to reduce computational cost. We propose Partition min-Hash (PmH), a novel hashing scheme to exploit locality, i.e. the fact that duplicate regions are usually localized in an image. In PmH, an image is first divided into overlapping partitions. Hashing is then applied independently to the visual words within each partition to compute a min-hash value.

Since the duplicate regions are localized, it is likely that one of the overlapping partition contains the common region among partial-duplicate images. By hashing within the partition instead of over all of the image, the min-hash is more repeatable among partial-duplicate images. By theoretical analysis, simulation, and experiments on real images, we show that PmH not only outperforms standard min-Hash in both precision and recall, but is also more than ten times faster for hashing, and more than two times faster overall including image preprocessing (at 1000 sketches/image). We also show that, when allotted the same amount of time, the proposed method achieves much higher recall than previous methods.

Partition min-hash and geometric min-hash can be used in conjunction by first partitioning images and then applying GmH to each partition. This improves the precision/recall of GmH, while speeding-up hashing by an order of magnitude.

To further utilize geometric constraints among visual words, we augment PmH by encoding the geometric structure in the sketches. Specifically, the geometric relationship among visual words in a sketch is quantized into an ID. This ID is then concatenated to the sketch to form the final representation of the image region.

## 1.1 Related Work

Large scale partial duplicate image discovery is closely related to image retrieval, which includes two popular themes. One theme represents an image as a bag of visual words, and then applies approaches from the text domain for efficient image indexing and retrieval [5,6,7,8]. Another theme uses hashing schemes to efficiently find similar images [3,9,10,11,12].

Naive application of image retrieval methods to partial duplicate image discovery can be done by using every image in the set as a query image. This has the computational complexity of the retrieval method multiplied by the number of images, which becomes prohibitive if the computational complexity of the retrieval method is more than  $O(1)$ . Hashing based methods are more suitable for partial duplicate image discovery, because all images can be hashed into a hash table and hash collisions can be retrieved as similar images, which can then be further expanded into more complete image clusters by image retrieval [4].

In this paper, we focus on designing efficient hashing schemes for scalable partial duplicate image discovery. Like previous works, we represent an image as a set of visual words [5], which are obtained by quantizing local SIFT feature descriptors [13,14]. Min-hash [1] and its variants can then be applied to finding similar sets and therefore similar images [3,4]. In particular, we are inspired by geometric min-hash [4].

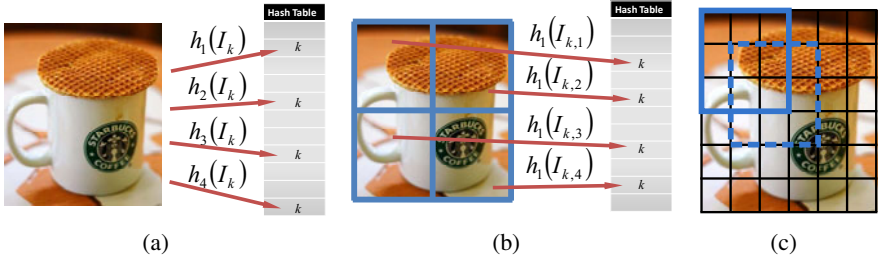
## 2 Min-Hash Algorithm

In this section we present some background on the min-hash algorithm. Min-hash is a Locality Sensitive Hashing scheme [15] that approximates similarity between sets. When an image is represented as a set of visual words, the similarity between two images can be defined as the Jaccard similarity between the two corresponding sets of visual words  $I_1$  and  $I_2$ :

$$\text{sim}(I_1, I_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|},$$

which is simply the ratio of the intersection to the union of the two sets.

Min-hash is a hash function  $h : I \mapsto v$ , which maps a set  $I$  to some value  $v$ . More specifically, a hash function is applied to each visual word in the set  $I$ , and the visual word that has minimum hashed value is returned as the min-hash  $h(I)$ . One way to implement the hash function is by a look-up table, with a random floating-point value assigned for each visual word in the vocabulary, followed by a min operator. The computation of the min-hash of a set  $I$  involves computing a hash of every element in the set and the time taken is therefore linear in the size of the set  $|I|$ . More details on min-hash can be found in [13].



**Fig. 2.** (a) In standard min-hash, min-hash sketches are extracted from the entire image. (b) In Partition min-hash, the image is divided into partitions and a min-hash sketch is extracted for each partition. (c) Overlapping partitions (in thick blue solid/broken line) are more likely to capture the entire duplicate region and lead to better performance. Sketches can be pre-computed for each grid element (thin black line) to avoid most of the redundant computation for overlapping partitions.

Min-hash has the property that the probability of hashing collision of two sets is equal to their Jaccard similarity:

$$P(h(I_1) = h(I_2)) = sim(I_1, I_2).$$

Since the output of a min-hash function  $v$  is actually a visual word, it carries the position and scale information of the underlying local feature for geometric verification.

In image retrieval or partial duplicate image discovery, we are interested in finding images which have similarity greater than some threshold  $\theta$ . In other words, we would like the probability of collision to be a step function:

$$P(h(I_1) = h(I_2)) = \begin{cases} 1 & \text{if } sim(I_1, I_2) \geq \theta; \\ 0 & \text{otherwise.} \end{cases}$$

This step function can be approximated by applying  $k$  min-hashes to a set and concatenating them into a *sketch*. Then  $n$  sketches can be computed for an image and all of them can be added to the hash table. Under this setting, two images will collide if they share one common identical sketch. The probability for two images to collide in the hash table becomes:

$$P(h(I_1) = h(I_2)) = 1 - (1 - sim(I_1, I_2)^k)^n, \tag{1}$$

which approximates the step function. The sharpness of the “step” and threshold  $\theta$  can be controlled by varying the sketch size  $k$  and the number of sketches  $n$ .

This scheme of computing  $n$  min-hash sketches of size  $k$  will be the baseline for our method, and we denote it as “standard min-hash”.

### 3 Partition Min-Hash

Based on the observation that duplicate regions among partial-duplicate images are usually localized, we propose a new method called Partition Min-Hash. It has better precision and recall and runs orders of magnitude faster than standard min-hash. It is also



**Algorithm 1.** Partition min-hash

---

```

Initialize  $N_s$  independent min-hash sketch functions  $h_i$ , where  $i = 1, \dots, N_s$ .
Initialize  $N_s$  hash tables  $T_i$ , which map a min-hash sketch  $s$  to image ID  $k$ .
for all Images  $I_k$  in database do
  Divide image  $I_k$  into a grid,  $I_{k,j}$ , where  $j = 1, \dots, N_g$ 
  For each partition, determine the grid elements that are associated with the partition.
  for  $i = 1, \dots, N_s$  do
    for  $j = 1, \dots, N_g$  do
      Extract sketch  $s_i \leftarrow h_i(I_{k,j})$ 
    for all Partitions do
      Look up sketches  $s_j$  extracted from grids that belong to current partition, and select
      true min-hash sketch  $s^*$ 
      Add  $(s^*, k)$  to hash table  $T_i$ 

```

---

very easy to implement and only has partition size and overlap as tuning parameters. The rest of this section introduces the method and discusses its performance.

### 3.1 Method Details

An image is divided into  $p$  rectangular regions of equal area, which we call partitions (Figure 2(b)). Then, instead of extracting min-hash sketches from the entire image, min-hash sketches are extracted for each partition independently. The  $p$  min-hash sketches, one extracted from each partition, are inserted into the hash table.

As will be analyzed in the following section, partitions that fully and tightly capture a duplicate region between images lead to better precision and recall, compared to cases in which a duplicate region spans several partitions, or where the partitions are much larger than the duplicate region. With evenly divided partitions, the duplicate is often split into two or more partitions. To alleviate this, we design partitions to be overlapping (Figure 2(c)) and multi-scale. This gives us a better likelihood of having at least one partition that captures the duplicate region completely. This may remind the reader of the sliding window technique, widely used for object detection. The spirit is, in fact, similar: we are hoping for one of the subwindows to hit a region of interest, which, in our case, is an unknown duplicate region.

We can avoid redundant computation on overlapping partitions by precomputing and reusing min-hashes. An image is divided into a grid, where the grid elements are the greatest common regions among partitions that cover that region (Figure 2(c)). Min-hash sketches are precomputed for each grid element  $w_i$ . Then the min-hash sketch for a partition  $P$  is computed by looking up elements  $\{w_i\}$  that are associated with that partition  $P$  and picking the true min-hash sketch among the precomputed min-hash sketches on elements:

$$h(P) = \min_i \{h(w_i) | w_i \in P\}$$

The entire algorithm is summarized in Algorithm 1.

### 3.2 Theoretical Analysis on Performance

In this section, we will analyze the speed and performance of partition min-hash and show that it achieves higher precision and recall in less time than standard min-hash.

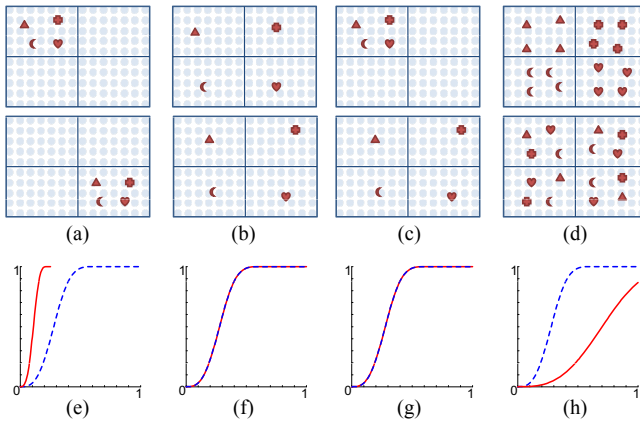
For the sake of comparison, we will keep the number of sketches per image equal for both cases. With the same number of sketches stored in the hash table, the two methods will use the same amount of memory. For example, if  $n$  sketches per image were computed for standard min-hash, we will compute  $n/p$  sketches for each of the  $p$  partitions for partition min-hash, so that the total number of sketches per image equals  $p \cdot n/p = n$ .

**Speed.** Processing time can be divided into two components: the time to compute sketches and the time for other overhead operations, such as reading the image from disk, extracting visual words, adding sketches to hash tables, etc. The overhead will be the same for both methods, with a fixed number of images and sketches per image for both methods, so the difference in time will be in computing sketches.

As mentioned in Section 2 the time to compute a min-hash sketch is linear in the size of the set. In partition min-hash, each sketch is the result of applying min-hash to a partition instead of the entire image, so the time it takes to compute each sketch is reduced by a factor of  $M_i/M$ , where  $M_i$  is the number of features in partition  $i$  and  $M$  is the number of features in image. On average,  $M_i/M$  will be roughly equal to the ratio of the area of the partition and the area of the image, which, in the case of non-overlapping partitions, is  $1/p$ , where  $p$  is the number of partitions. So the overall time to compute sketches reduces by  $1/p$ . In the case of overlapping partitions, additional time is required to look up grid elements associated with each partition, compared to the non-overlapping case, however this is small compared with the time to hash features. The key to this speedup is that in standard min-hash, each feature participates in all  $n$  sketches, so must be hashed  $n$  times. In PmH, each feature only participates in  $n/p$  sketches, so the number of computed hashes is reduced by a factor of  $p$ .

**Precision and Recall.** We now show that the sketches created by partition min-hash have better discriminative power than the sketches created by standard min-hash, despite the fact that they take less time to compute. We will study precision and recall by analyzing the collision probability of true matching image pairs and the collision probability of false matching image pairs. The collision probability of true matching pairs is equal to recall, defined as the number of retrieved positives divided by the number of true positives. The collision probability of false matching pairs is related to (but not equal to) precision, defined as the number of retrieved positives divided by the number of retrieved images. We have derived in Section 2 Equation (1) that the collision probability of standard min-hash is equal to  $P(h(I_1) = h(I_2)) = 1 - (1 - sim(I_1, I_2)^k)^n$ . We will show that partition min-hash achieves higher collision probability for true matching pairs and lower probability for false matching pairs, thus achieving higher precision and recall.

Let us first analyze the collision probability of true matching image pairs. To simplify the analysis, we will consider the case of non-overlapping partitions (Figure 2(b)). The arrangement of partitions with respect to the region with duplicate content can then be categorized into three cases, illustrated in Figure 3(a),(b),(c). For the purposes of analysis, we have assumed that features are spread across the image and each partition within an image contains the same number of features (including matching and non-matching background features). Once the simplified analysis on non-overlapping cases



**Fig. 3.** (a)(b)(c): Illustration of true matching image pairs. Various symbols in red represent matching features across images. Lightly colored circles in the background represent non-matching features and are assumed to be spread out uniformly across partitions. (a): The duplicate region is captured in a single partition in each image. (b): The duplicate region is split across all partitions. (c): Mix of (a) and (b). (d): Illustration of a false matching image pair. Features are randomly distributed. (e)(f)(g)(h): Collision probabilities plotted against image similarity for cases (a)(b)(c)(d), respectively. Red solid curve: Collision probability of partition min-hash. Blue broken curve: Collision probability of standard min-hash. X axis: Similarity. Y axis: Collision probability.

is done, it will be easy to infer that overlapping partitions are more likely to generate “preferred” partitions.

*Case (a):* The duplicate region is contained within a single partition in each image. Since duplicate features are contained in only 1 of the  $p$  partitions, the similarity between the two images  $sim(I_1, I_2)$  must be less than  $1/p$ . Now, the similarity between the partitions containing duplicate region is  $p \cdot sim(I_1, I_2)$ , where  $p$  is the number of partitions. Since  $n/p$  sketches are extracted from those partitions, the overall collision probability is equal to

$$P(h(I_1) = h(I_2)) = 1 - \left(1 - (p \cdot sim(I_1, I_2))^k\right)^{n/p},$$

and is plotted in Figure 3(e). Partition min-hash achieves a higher collision probability than standard min-hash in this case.

It is possible for duplicate regions to lie in between partitions. In such case, it can be shown that the collision probability is less than what was derived above, but is still greater than standard min-hash. Moreover, overlapping partitions increases the chance of having a partition that covers a duplicate region, thus increasing the collision probability even further. We obtained the collision probability for overlapping partitions through simulation and it is reported at the end of this section.

*Case (b):* The duplicate regions are split up among partitions. The illustration shows the most extreme case where the duplicate region is split across all  $p$  partitions. Since we are considering an actual duplicate image, each partition from one image will have a corresponding matching partition in the other image, e.g. partition 1 from image 1

matches with partition 1 from image 2, partition 2 from image 1 matches with partition 2 image 2, and so on. Now for each pair of matching partitions, the similarity between the pair will be the same as the original similarity  $sim(I_1, I_2)$ . For each partition,  $n/p$  sketches are extracted, but the image will be considered as colliding if any one of the  $p$  pairs collide. So the overall collision probability is equal to

$$\begin{aligned} P(h(I_1) = h(I_2)) &= 1 - \left( \left( 1 - (sim(I_1, I_2))^k \right)^{n/p} \right)^p \\ &= 1 - (1 - sim(I_1, I_2)^k)^n, \end{aligned}$$

which is the same as the collision probability for standard min-hash, and it is plotted in Figure 3(f). In practice, the splitting of duplicate regions will typically not be as extreme as a split across all  $p$  partitions and the collision probability will be somewhere in between case (a) and case (b).

*Case (c):* The duplicate region is contained in one partition in one image and split up into  $p$  partitions in the other image. The partition which contains the entire duplicate region has non empty intersection with all  $p$  partitions from the other image and has the probability to have the same min-hash value proportional to their similarity, which is equal to  $sim(I_1, I_2)$ , as on average the number of duplicate features and the number of non-duplicate features are reduced by the same ratio from the entire image. Again in this case, the collision probability is equal to

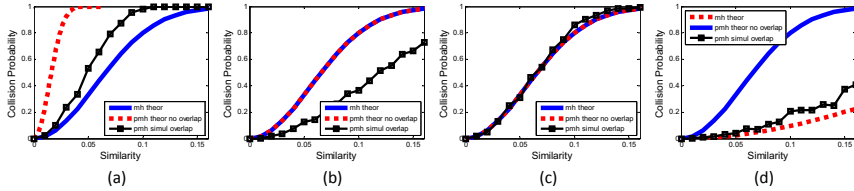
$$P(h(I_1) = h(I_2)) = 1 - (1 - sim(I_1, I_2)^k)^n,$$

plotted in Figure 3(g), but in practice most images will lie somewhere between cases (a) and (c). The simulation at the end of the section confirms the above analytic result.

*Case (d):* The collision probability of false matching image pairs. We assume that in a false match, duplicate features among two images are scattered randomly across the image, as opposed to being localized in some partition. This is illustrated in Figure 3(d). Partitioning randomly scattered features can be considered as random sampling, and the expected similarity between one randomly sampled partition and another randomly sampled partition reduces by a factor of  $p$ , i.e., the expected similarity between any partitions in the two images is equal to  $sim(I_1, I_2)/p$ . As there are  $p$  partitions for each image, there are a total of  $p^2$  combinations leading two partitions to collide. Therefore, the collision probability is equal to

$$\begin{aligned} P(h(I_1) = h(I_2)) &= 1 - \left( \left( 1 - (sim(I_1, I_2)/p)^k \right)^{n/p} \right)^{p^2} \\ &= 1 - (1 - (sim(I_1, I_2)/p)^k)^{np}, \end{aligned}$$

which is lower than standard min-hash, and is plotted in Figure 3(h). In practice, some partitions will have higher number of duplicate features and have higher similarity than  $sim(I_1, I_2)/p$ , which leads to an overall collision probability that is higher than what is derived above. But the chances of having a partition with significantly high number of duplicate features will be low, and the true collision probability will be close to what we derived.



**Fig. 4.** Simulated collision probability of partition min-hash with overlapping partitions for four cases. ‘mh theor’: Theoretical rate of standard min-hash. ‘pmh theor no overlap’: Theoretical rate of partition min-hash with non-overlapping partitions. ‘pmh simul overlap’: Simulated rate of partition min-hash with overlapping partitions.

*Overlapping partitions and simulation verification.* So far, the analysis has been done for non-overlapping partitions where the duplicate region is also within some partition. In practice, the duplicate region may stride over partition boundaries, so we use overlapping partitions to achieve higher chance of capturing the duplicate region in one partition. The theoretical collision probability for overlapping partitions is complicated due to dependence among sketches from overlapping partitions. Instead, we use synthetic examples to simulate the case where the duplicate regions are not aligned with partitions for the four cases in Fig. 3 and apply partition min-hash with overlapping partitions. The simulation was done by synthesizing images with visual words distributed as described in the four cases and applying partition min-hash to the images. The simulation is repeated 1000 times and the collision probabilities are reported in Figure 4.

Compared to standard min-hash, we see that the collision probability of partition min-hash with overlapping partitions is higher in case (a) and similar in case (c). For case (d), which is false matching, the probability of false collision is much lower than standard min-hash. Compared to the ideal non-overlapping case, both (c) and (d) have similar performance, while (a) is not as good as the ideal case. This is expected as we are using the same number of sketches for both overlapping and non-overlapping cases. As a result, the number of sketches per partition for overlapping case is lower. This affects case (b) the most where its collision probability is lower than the standard min-hash. In practice, most duplicates will occupy a portion of the image and will be in between case (a) and case (b). Moreover, since we pre-computed min-hash for each grid, we can get more sketches for each overlapping partition almost for free.

## 4 Evaluation of Partition Min-Hash

In this section, we present quantitative evaluations of our method using two datasets, our own dataset collected from the web and the Oxford buildings dataset [6].

### 4.1 Experimental Setup

In our own dataset, we have collected 778 images from the web and manually categorized them into 19 categories, where the images within each category are partial duplicates. There are no exact duplicates among this set of 778 images. The set contains 17595 true

matching image pairs (belonging to the same category) out of  $778 \times 777/2 = 302253$  total pairs. Such a large set of image pairs is adequate for evaluating hashing schemes. The average Jaccard similarity for true pairs is 0.019. The number of features per image ranges from 200 to 1000, and we have quantized them using a visual word vocabulary with one million visual words. It takes about 100ms per image to extract visual words. Min-hash functions are implemented as lookup tables of random floating point values assigned per each visual word, followed by a min operation.

The Oxford buildings dataset [6] consists of 5062 images collected from Flickr by searching for particular Oxford landmarks. The annotation provides whether an image in the database is a partial duplicate of a query image for 55 queries. As the time that was taken to extract visual words were not reported in the Oxford buildings set, we assumed it took 100ms per image in our reported graphs.

We tested the quality of collision pairs that are retrieved by counting the number of true pairs and false pairs that were retrieved using our proposed method. Recall was computed as the number of retrieved true *pairs* divided by the total number of true pairs. Precision was computed as the percentage of true *pairs* among all retrieved pairs. This measure differs from the number of *images* retrieved from a set of partial duplicate images. F-measure was computed as the harmonic mean of precision and recall.

In order to get the final clustered set of duplicate images, post-operations, such as connected components and query expansion, should be performed, since min-hash based methods only retrieve a subset of pairs of images in a group of duplicate images probabilistically and does not complete the clustering. We did not include the post-operations in our evaluation and evaluated only the pairs that it retrieved, as it allows a more direct comparison of the various min-hash methods themselves, which is the focus of our study.

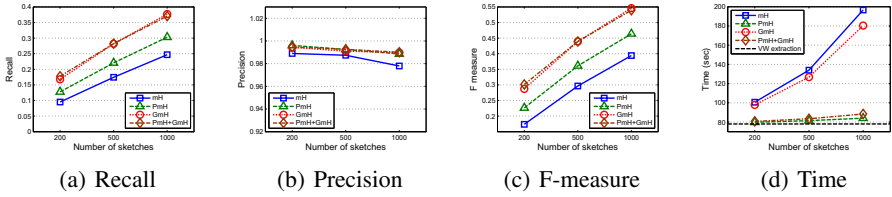
Experiments were run on a single 32-bit machine with a 3.2 GHz processor and 4GB memory.

## 4.2 Results on Our Dataset

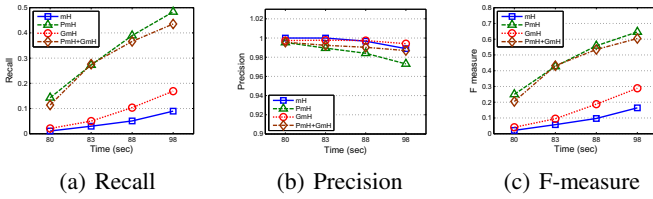
We have empirically tested the effect the number of partitions per image and the area of overlap of neighboring partitions, by varying them from 16 to 144 and 0% to 80%, respectively. An overlap area of 50% gave the best recall. Recall tends to increase as the number of partitions increase. The time taken was the least between 64 and 121 partitions. We have chosen 100 partitions per image and 50% overlap for the following experiments. We have used two hashes per sketch in our experiments.

We have tested and compared four methods: standard min-hash (mH), partition min-hash (PmH), geometric min-hash (GmH) [4], and the combination of min-hash and geometric min-hash (PmH+GmH). They were compared under two scenarios: constant number of sketches per image (Figure 5) and constant runtime (Figure 6). The first scenario allows us to evaluate how discriminative the sketches are for each method, and how long it takes to compute a single sketch. The second scenario allows us to evaluate how our proposed method compares given the same computational resource.

In the first scenario (Figure 5), all hashing schemes have high precision, with PmH being slightly better than mH. In the mean time, PmH improves the recall by more than 20% when compared to mH. The speed of the hashing process of PmH is 16 times



**Fig. 5.** Performance on our dataset with fixed number of sketches per image. F-measure is the harmonic mean of precision and recall. Time scale starts around the time it took for extracting visual words from images, denoted by “VW extraction.”

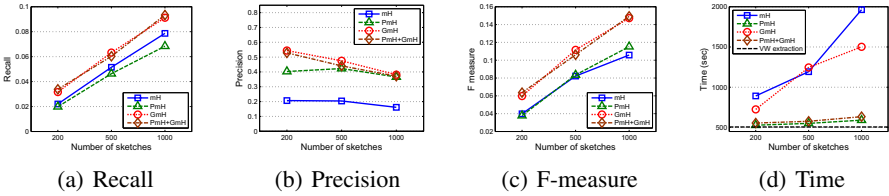


**Fig. 6.** Performance on our dataset with fixed time budget

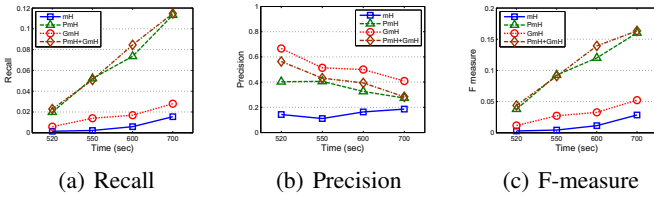
faster than mH. When our partition scheme is applied to GmH, speed improves by 9 times. When the time for extracting visual words is added, PmH is 2.5 times faster than mH, and PmH+GmH is 2 times faster than GmH, at 1000 sketches/image. At 1000 sketches/image and 2 min-hash/sketch and assuming 1000 features per image, min-hash requires about two million table look up operations, which is a significant amount of computation. Partition min-hash reduces this number of look up operations by a factor proportional to the number of partitions. Furthermore, the overall improvement in speed becomes more significant when more sketches per image ( $n$ ) and more min-hashes per sketch ( $k$ ) are used and min-hash operation contributes more to the overall execution time. It is beneficial to use a greater number of sketches and min-hashes, because it approximates the ideal step function better, which was discussed in Section 2, and leads to better performance. The constraint that limits the number of sketches and min-hashes is the computation time, and partition min-hash alleviates this constraint.

In the second scenario (Figure 6), PmH and PmH+GmH have much higher recall than mH and GmH. When allowed to run for 5 seconds, our PmH has 9.1 times higher recall than mH, and PmH+GmH has 9.2 times higher recall than mH, while GmH has 1.7 times higher recall than mH (mH: 3.0%, PmH: 27.2%, GmH: 5.0%, PmH+GmH: 27.6%). All of the hashing schemes have high precision (mH: 100%, PmH: 98.9%, GmH: 99.7%, PmH+GmH: 99.2%). As we can see, given a fixed time budget in real applications, the speed up of our partition min-hash leads to significant improvement in recall.

Figure 9 shows a sample set of images having min-hash sketch collisions using PmH+ GmH with 500 sketches and 100 partitions per image with 50% overlap. In a typical application these collisions are used as seeds to complete the image clusters using query expansion [4].



**Fig. 7.** Performance on Oxford buildings dataset with fixed number of sketches per image. Time scale starts around the time it took for extracting visual words from images, denoted by “VW extraction.” (100ms per image was assumed)



**Fig. 8.** Performance on Oxford buildings dataset with fixed time budget

**4.3 Results on Oxford Buildings Dataset**

We have performed the same experiments on the Oxford building dataset. Figure 7 shows results with fixed number of sketches per image. On this dataset, the performance of min-hash is low, with a particularly low precision of about 20%. With such low precision, the improvement made by Partition min-hash is more pronounced—the precision improvement over mH is 200%. The speed improvement is also significant, consistent with our own dataset. For hashing, PmH runs 17 times faster than mH, and PmH+GmH runs 11 times faster than mH. Our approach also speeds up GmH by 7 times for hashing, without losing precision or recall. When time for extracting visual words is added, PmH is 3.3 times faster than mH, and PmH+GmH is 2.4 times faster than GmH, at 1000 sketches/image.

Figure 8 shows results with fixed runtime. With the same amount of computational resource, PmH and PmH+GmH achieves significant improvement in recall (mH:0.6%, PmH: 7.4%, GmH: 2.3%, PmH+GmH: 8.5%, Time: 100sec).

**4.4 Scalability On Six Million Images**

To demonstrate the scalability of our method, we applied PmH+GmH to search for all partial-duplicate matches in a dataset of six million images collected from the web. The method took 131 minutes to run on a single 3.2GHz machine with 4GB memory, with the following parameters: 16 partitions per image with 50% overlap and 16 sketches per image. Our method was able to retrieve many partial duplicate images, however, since we have no ground-truth available for this image corpus, we do not present quantitative results other than timing information.



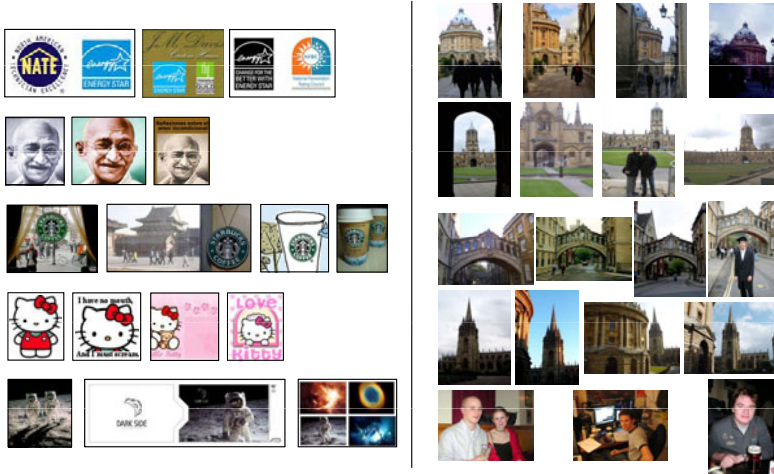


Fig. 9. Example images with sketch collisions. Left: Our dataset. Right: Oxford buildings dataset

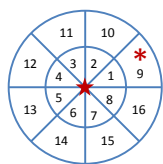
## 5 Geometry Sensitive Hash

A typical application of minHash is to use the collisions as cluster seeds that will be expanded (by image retrieval) into complete clusters of partial-duplicate images [4]. In doing so, it is important to reduce false positives in these seeds before they are verified by full geometric verification [4], especially for large scale data set where the number of false positives tends to increase.

The local geometric structure of features in duplicate regions is usually preserved across images. For example, in Figure 11, the top of the Eiffel tower is always above the base, and the word “Starbucks” is always above the word “coffee”. Instead of verifying these local geometric relationships after sketch collisions are retrieved, we encode such local geometric structure into the sketches, so that they can be checked at an earlier stage. This reduces the number of false positive collisions, and therefore reduces the number of full geometric verifications that need to be performed after expansion by image retrieval, saving computational expense.

We encode geometric structure into the sketches by hashing the geometric relationship between features. This is achieved by creating an integer ID which encodes the relative geometric configuration<sup>1</sup> among visual words in a sketch, and concatenating it to the min-hash sketch. We call this Geometry Sensitive Hashing. There are many ways to hash the local geometric structure using the relative location and scale of features. We use a simple hash function to quantize the geometric structure into 32 IDs, 16 for the relative position of two features (2 along the radial direction (near, far), and 8 along the tangential direction), and 2 for relative scale (Fig. 10(a)). When there are more than two features in a sketch, hashes for all combination of pairs are concatenated.

<sup>1</sup> The scale and dominant orientation output by feature detectors can be used to normalize the coordinate system at each point to derive the relative configuration.



(a)

	Without GSH			With GSH		
	TP	FP	Time (s)	TP	FP	Time (s)
<b>Standard</b>	3132	40	56.0	2731	9	55.8
<b>PmH</b>	3963	30	3.4	3617	9	3.5
<b>GmH</b>	5065	46	48.8	4778	32	48.9
<b>PmH+GmH</b>	5066	38	5.4	4744	26	5.5

(b)

**Fig. 10.** (a) Geometry sensitive hashing (sketch size = 2): a grid is defined centered at the first visual word in the sketch. The second visual word \* has a grid id of 9, which is used as part of the hash key. (b) Evaluation of GSH. TP/FP: number of retrieved true/false positive pairs.

**Evaluation of Geometry-Sensitive Hashing.** Figure 10(b) shows the number of true/false positives when applying Geometry-Sensitive Hashing(GSH), given 500 sketches per image. It shows that GSH decreases the number of false positives for all 4 hashing schemes with a negligible computational overhead. We have also observed that GSH is more effective for PmH than for GmH in reducing the number of false positives.

## 6 Conclusion

We have proposed two novel improvements to min-hash for discovering partial duplicate images in a large set of images: partition min-hash and geometry-sensitive hash. They are improved hashing functions which make use of the geometric configuration information available in images, and take advantage of the fact that duplicate regions are localized in an image and that geometric configurations are preserved across duplicate regions. The methods are easy to implement, with few tuning parameters. We have shown that the proposed hashing method achieves higher precision and recall and runs orders of magnitude faster than the current state-of-the-art. We have also shown that the speed-up allows us to afford a larger number of sketches, which in turn improves the hashing performance, given the same amount of computational resource. Although we have shown the effectiveness of partition min-hash in the domain of images, this method may be applicable to other domains where min-hash is used, such as duplicate document detection, if similar locality properties exist in those domains.

## References

1. Broder, A.Z.: On the resemblance and containment of documents. In: Compression and Complexity of Sequences SEQUENCES'97 (1997)
2. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: Proc. of the Int. Conf. on Image and Video Retrieval (2007)
3. Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-hash and tf-idf weighting. In: Proceedings of the British Machine Vision Conference (2008)
4. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: CVPR (2009)
5. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)

6. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
7. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
8. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicateweb image search. In: CVPR (2009)
9. Grauman, K., Darrell, T.: Pyramid match hashing: Sub-linear time indexing over partial correspondences. In: CVPR (2007)
10. Jain, P., Kulis, B., Grauman, K.: Fast image search for learned metrics. In: CVPR (2008)
11. Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and sub-image retrieval. In: Proc. of ACM Int. Conf. on Multimedia (2004)
12. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large databases for recognition. In: IEEE CVPR (2008)
13. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 20, 91–110 (2003)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27(10), 1615–1630 (2005)
15. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proc. of ACM symposium on Theory of computing (1998)

# Automatic Attribute Discovery and Characterization from Noisy Web Data

Tamara L. Berg<sup>1</sup>, Alexander C. Berg<sup>2</sup>, and Jonathan Shih<sup>3</sup>

<sup>1</sup> Stony Brook University, Stony Brook NY 11794, USA  
tlberg@cs.sunysb.edu

<sup>2</sup> Columbia University, New York NY 10027, USA  
aberg@cs.columbia.edu

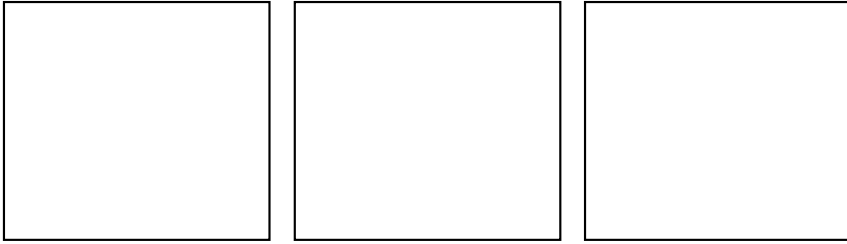
<sup>3</sup> University of California, Berkeley, Berkeley CA 94720, USA  
jmshih@berkeley.edu

**Abstract.** It is common to use domain specific terminology – attributes – to describe the visual appearance of objects. In order to scale the use of these describable visual attributes to a large number of categories, especially those not well studied by psychologists or linguists, it will be necessary to find alternative techniques for identifying attribute vocabularies and for learning to recognize attributes without hand labeled training data. We demonstrate that it is possible to accomplish both these tasks automatically by mining text and image data sampled from the Internet. The proposed approach also characterizes attributes according to their visual representation: global or local, and type: color, texture, or shape. This work focuses on discovering attributes and their visual appearance, and is as agnostic as possible about the textual description.

## 1 Introduction

Recognizing attributes of objects in images can improve object recognition and classification as well as provide useful information for organizing collections of images. As an example, recent work on face recognition has shown that the output of classifiers trained to recognize attributes of faces – gender, race, etc. – can improve face verification and search [1, 2]. Other work has demonstrated recognition of unseen categories of objects from their description in terms of attributes, even with *no* training images of the new categories [3, 4] – although labeled training data is used to learn the attribute appearances. In all of this previous work, the sets of attributes used are either constructed *ad hoc* or taken from an application appropriate ontology. In order to scale the use of attributes to a large number of categories, especially those not well studied by psychologists or linguists, it will be necessary to find alternative techniques for identifying attribute vocabularies and for learning to recognize these attributes without hand labeled data.

This paper explores automatic discovery of attribute vocabularies and learning visual representations from unlabeled image and text data on the web. For



**Fig. 1.** Original input image (left), Predicted attribute classification (center), and Probability map for localization of “high,heel” (right – white indicates high probability)

example, our system makes it possible to start with a large number of images of shoes and their text descriptions from shopping websites and automatically learn that “stiletto” is a visual attribute of shoes that refers to the shape of a specific region of the shoe (see Fig 6). This particular example illustrates a potential difficulty of using a purely language based approach to this problem. The word “stiletto” is a noun that refers to a knife, except, of course, in the context of women’s shoes. There are many other examples, “hobo” can be a homeless person or a type of handbag (purse), “wedding” can be a visually distinctive (color!) feature of shoes, “clutch” is a verb, but also refers to a type of handbag. Such domain specific terminology is common and poses difficulties for identifying attribute vocabularies using a generic language based approach. We demonstrate that it is possible to make significant progress by analyzing the connection between text and images using almost no language specific analysis, with the understanding that a system exploiting language analysis in addition to our visual analysis would be a desirable future goal.

Our approach begins with a collection of images with associated text and ranks substrings of text by how well their occurrence can be predicted from visual features. This is different in several respects from the large body of work on the related problem of automatically building models for object category recognition [5, 6]. There, training images are labeled with the presence of an object, with the precise localization of the object or its parts left unknown. Two important differences are that, in that line of work, images are labeled with the name of an object category by hand. For our experiments on data from shopping websites, images are not hand curated for computer vision. For example, we do know the images of handbags in fact contain handbags with no background clutter, but the text to image relationship is significantly less controlled than the label to image relationship in other work – *e.g.*, it is quite likely that an image showing a black shoe will not contain the word “black” in its description. Furthermore, there are a range of different terms that refer to the same visual attribute (*e.g.*, “ankle strap” and “strappy”). Finally, much of the text associated with the images does not in fact describe any visual aspect of the object (see Fig. 2). We must identify the wheat from amidst a great deal of chaff.

More related to our work are a series of papers modeling the connection between words and pictures [7–10]. These address learning the relationships between text and images at a range of levels – including learning text labels associated with specific regions of images. Our focus is somewhat different, learning vocabularies of attributes for particular object categories, as well as models for the visual depiction of these attributes. This is done starting with more free-form text data than that in corel [7] or art catalogues [8]. We use discriminative instead of generative machine learning techniques. Also this work introduces the goal of ranking attributes by visualness as well as exploring ideas of attribute characterization.

The process by which we identify which text terms are visually recognizable tells us what type of appearance features were used to recognize the attribute – shape, color, or texture. Furthermore, in order to determine if attributes are localized on objects, we train classifiers based on local sets of features. As a result, we can not only rank attributes by how visually recognizable they are, but also determine whether they are based on shape, color, or texture features, and whether they are localized – referring to a specific part of an object, or global – referring to the entire object.

**Our contributions are:**

1. Automatic discovery and ranking of visual attributes for specific types of objects.
2. Automatic learning of appearance models for attributes without any hand labeled data.
3. Automatic characterization of attributes on two axes: a) the relevant appearance features – shape, color, or texture, b) localizability – localizable or global.

**Approach:**

Our approach starts with collecting images and associated text descriptions from the web (Sec 4.1). A set of strings from the text are considered as possible attributes and ranked by visualness (Sec 2). Highly ranked attributes are then characterized by feature type and localizability (Sec 3.1). Performance is evaluated qualitatively, quantitatively, and using human evaluations (Sec 4).

**1.1 Related Work**

Our key contribution is automatic discovery of visual attributes and the text strings that identify them. There has been related work on using hand labeled training data to learn models for a predetermined list (either formed by hand or produced from an available application specific ontology) of attributes [1–4]. Recent work moves toward automating part of the attribute learning process, but is focused on the constrained setting of butterfly field guides and uses hand coded visual features specific to that setting, language templates, and predefined attribute lists (lists of color terms etc) [11] to obtain visual representations from text alone. Our goal is instead to automatically identify an attribute vocabulary and their visual representations without the use of any prior knowledge.



Dazzle after dark with Judith Leiber's decadent oversized crystal-embellished silver-tone clutch. Carry this fabulous extra to add high-octane glamour to an LBD and teetering heels. Shown here with an Emilio Pucci dress and Givenchy shoes.



The 12K pink and green gold leaves gently cascade down on these delicate beaded 10K gold earrings.



Rock and roll in these sexy, strappy heels from Report Signature. The smoldering Rockwell features a grey patent leather upper with pleated satin crossing at the open-toe atop a 1 inch platform, patent straps closing around the ankle with a gold buckled, and finally a 5 inch patent cone heel. Sizzle in these fierce mile-high shoes.

**Fig. 2.** Example input data (images and associated textual descriptions). Notice that the textual descriptions are general web text, unconstrained and quite noisy, but often provide nice visual descriptions of the associated product.

Our discovery process identifies text phrases that can be consistently predicted from some aspect of visual appearance. Work from Barnard et al, *e.g.* [9], has looked at estimating the visualness of text terms by examining the results of web image search using those terms. Ferrari and Zisserman learn visual models of given attributes (striped, red, etc) using web image search for those terms as training data [12]. Other work has automatically associated tags for photographs (in Corel) with segments of images [7]. Our work focuses on identifying an attribute vocabulary used to describe specific object categories (instead of more general images driven by text based web search for a given set of terms) and characterizes attributes by relevant feature types and localizability.

As mentioned before, approaches for learning models of attributes can be similar to approaches for learning models of objects. These include the very well known work on the constellation model [5, 6], where images were labeled with the presence of an object, but the precise localization of the object and its parts were unknown. Variations of this type of weakly supervised training data range from small amounts of uncertainty in precise part locations when learning pedestrian detectors from bounding boxes around whole a figure [13] to large amounts of uncertainty for the location of an object in an image [5, 6, 14, 15]. At an extreme, some work looks at automatically identifying object categories from large numbers of images showing those categories with no per image labels [16, 17], but even here, the *set* of images is chosen for the task. Our experiments are also close work on automatic dataset construction [18–22], that exploits the connection between text and images to collect datasets, cleaning up the noisy “labeling” of images by their associated text. We start with data for particular categories, rank attributes by visualness, and then go into a more detailed learning process to identify the appropriate feature type and localizability of attributes using the multiple instance learning and boosting (MILboost) framework introduced by Viola [23].



**Fig. 3.** Example input images for 2 potential attribute phrases (“hoops”, and “navy”). On the left of each pair (a,c) we show randomly sampled images that have the attribute word in their description. On the right of each pair (b,d) we show randomly sampled images that do not have the attribute word in their description. Note that these labels are very noisy – images that show “hoops” may not contain the phrase in their description, images described as “navy” may not depict navy ties.

## 2 Predicting Visualness

We start by considering a large number of strings as potential attributes (further described in Sec. 4) – for instance any string that occurs frequently in the data set can be considered. A visual classifier is trained to recognize images whose associated text contains the potential attribute. The potential attributes are then ranked by their average precision on held out data.

For training a classifier for potential attribute with text representation  $X$ , we use as positive examples those images where  $X$  appears in its description, and randomly sample negative examples from those images where  $X$  does not appear in the description. There is a fair amount of noise in this labeling (described in Section 4, see fig 3 for examples), but overall for good visual attribute strings there is a reasonable signal for learning. Because of the presence of noise in the labels and the possibility of overfitting, we evaluate accuracy on a held out validation set – again, all of the “labels” come directly from the associated, noisy, web text with no hand intervention.

We then rank the potential attributes by visualness using the learned classifiers by measuring average labeling precision on the validation data. Because boosting has been shown to produce accurate classifiers with good generalization, and because a modification of this method will be useful later for our localizability measure, we use AnyBoost on decision stumps as our classification scheme. Whole image based features are used as input to boosting (Sec 4.2 describes the low level visual feature representations).

### 2.1 Finding Visual Synsets

The web data for an object category is created on and collected from a variety of internet sources (websites with different authors). Therefore, there may be several attribute phrases that describe a single visual attribute. For example, “Peep-Toe”



and “Open-Toe” might be used by different sources to describe the same visual appearance characteristic of a shoe. Each of these attribute phrases may (correctly) be identified as a good visual attribute, but their resulting attribute classifiers might have very similar behavior when applied to images. Therefore, using both as attributes would be redundant.

Ideally, we would like to find a comprehensive, but also compact collection of visual attributes for describing and classifying an object class. To do so we use estimates of Mutual Information to measure the information provided by a collection of attributes to determine whether a new attribute provides significantly different information than the current collection, or is redundant and can therefore might be considered a synonym for one of the attributes already in the collection. We refer to a set of redundant attributes providing the same visual information as a *visual synset* of cognitive visual synonyms. To build a collection of attributes, we iteratively consider adding attributes to the collection in order by visualness. They are added provided that they provide significantly more mutual information for their text labels than any of the attributes already in the set. Otherwise we assign the attribute to the synset of the attribute currently in the collection that provided the most mutual information. This process results in a collection of attribute synsets that cover the data well, but tend not to be visually repetitive.

#### Example Shoe Synsets

```
{“sandal style round”, “sandal style round open”, “dress sandal”, “metallic”}
  {“stiletto”, “stiletto heel”, “sexy”, “traction”, “fabulous”, “styling”}
{“running shoes”, “synthetic mesh”, “mesh”, “stability”, “nubuck”, “molded”...}
  {“wedding”, “matching”, “satin”, “cute” }
```

#### Example Handbag Synsets

```
{“hobo”, “handbags”, “top.zip.closure”, “shoulder,bag”, “hobo,bag” }
  {“tote”, “handles”, “straps”, “lined”, “open”...}
  {“mesh”, “interior”, “metal”}
  {“silver”, “metallic” }
```

Alternatively, one could try to merge attribute strings based on text analysis – for example merging attributes with high co-occurrence or matching substrings. However, co-occurrence would be insufficient to merge all ways of describing a visual attribute, *e.g.*, “peep-toe” and “open-toe” are two alternative descriptions for the same visual attribute, but would rarely be observed in the same textual description. Matching substrings can lead to incorrect merges, *e.g.*, “peep-toe” and “closed-toe” share a substring, but have opposing meanings. Our method for visual attribute merging based on mutual information overcomes these issues.

### 3 Attribute Characterization

For those attributes predicted to be visual, we would like to make some further characterizations. To this end, we present methods to determine whether an attribute is localizable (Section 3.1) – ie does the attribute refer to a global appearance characteristic of the object or a more localized appearance characteristic? We also provide a way to identify attribute type (Section 3.2) – ie is the attribute indicated by a characteristic shape, color, or texture?



Fig. 4. Automatically discovered handbag attributes, sorted by visualness

### 3.1 Predicting Localizability

In order to determine whether an attribute is localizable – whether it usually corresponds to a particular part on an object, we use a technique based on MILBoost [14, 23] on local image regions of input images. If regions with high probability under the learned model are tightly clustered, we consider the attribute localizable. Figure 1 shows an example of the predicted probability map for the “high heel” attribute and our automatic attribute labeling.

MILBoost is a multiple instance learning technique using AnyBoost, first introduced in Viola et al [23] for face detectors, and later used for other object categories [14]. MILBoost builds a classifier by incrementally selecting a set of weak classifiers to maximize classification performance, re-weighting the training samples for each round of training. Because the text descriptions do not specify what portion of image is described by the attribute, we have a multiple instance learning problem where each image (data item) is treated as a bag of regions (samples) and a label is associated with each image rather than each region.

For image  $i$  and segment  $j$ , the boosted classifier predicts the score of a sample as a linear combination of weak classifiers:  $y_{ij} = \sum_t \lambda_t c^t(x_{ij})$ . Then the probability that segment  $j$  in image  $i$  is a positive example is:

$$p_{ij} = \frac{1}{1 + \exp(-y_{ij})} \quad (1)$$



Fig. 5. Automatically discovered shoe attributes, sorted by visualness

The probability of an image being positive is then (under the noisy OR model), one minus the probability of all segments being negative.

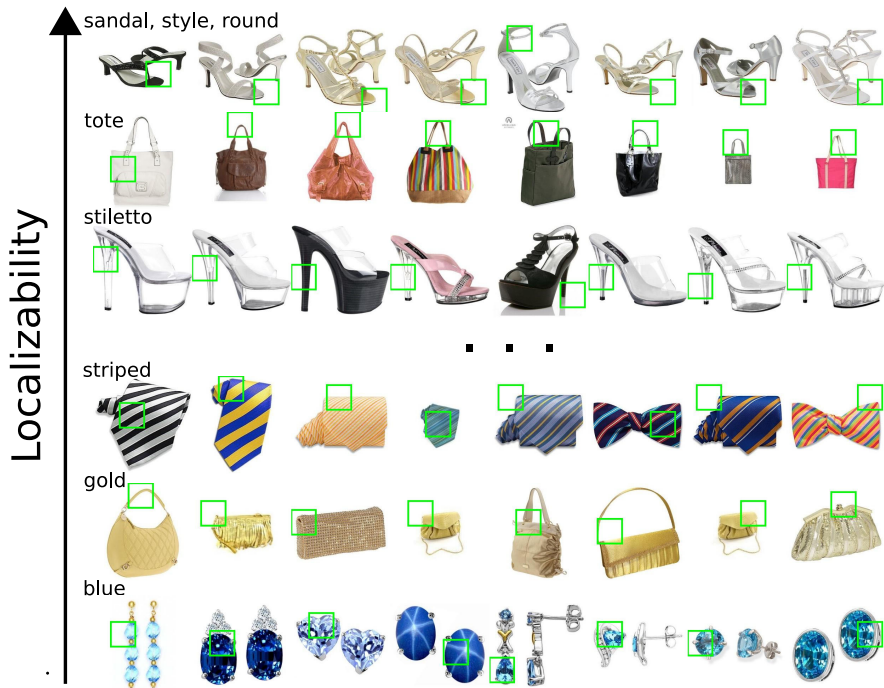
$$p_i = 1 - \prod_{j \in i} (1 - p_{ij}) \tag{2}$$

Following the AnyBoost technique, the weight,  $w_{ij}$ , assigned to each segment is the derivative of the cost function with respect to a change in the score of the segment (where  $t_i$  is the label of image  $i \in 0, 1$ ):

$$w_{ij} = \frac{t_i - p_i}{p_i} p_{ij} \tag{3}$$

Each round of boosting selects the weak classifier that maximizes:  $\sum_{ij} c(x_{ij})w_{ij}$ , where  $c(x_{ij}) \in \{-1, +1\}$  is the score assigned to the segment by the weak classifier. The weak classifier weight parameter,  $\lambda_t$  is determined using line search to maximize the log-likelihood of the new combined classifier at each iteration  $t$ .

**Localizability** of each attribute is then computed by evaluating the trained MILBoost classifier on a collection of images associated with the attribute. If the



**Fig. 6.** Attributes sorted by localizability. Green boxes show most probable region for an attribute.

classifier tends to give high probability to a few specific regions on the object (*i.e.*, only a small number of regions have large  $P_{ij}$ ), then the attribute is localizable. If the probability predicted by the model tends to be spread across the whole object then the attribute is a global characteristic of the object. To measure the attribute spread, we accumulate the predicted attribute probabilities over many images of the object and measure the localizability as the portion of image needed to capture the bulk of this accumulated probability (the portion of all  $P_{ij}$ 's containing at least 95% of the predicted probability). If this is a small percentage of the image then we predict the attribute as localizable. For our current system, we have focused on product images which tend to be depicted from a relatively small set of possible viewpoints (shoe pointed left, two shoes etc). This means that we can reliably measure localization on a rough fixed grid across the images. For more general depictions, an initial step of alignment or pose clustering [24] could be used before computing the localizability measure.

### 3.2 Predicting Attribute Type

Our second attribute characterization classifies each visual attribute as one of 3 types: color, texture and shape. Previous work has concentrated mostly on building models to recognize color (*e.g.*, “blue”) and texture (*e.g.*, “spotty”)

based attributes. We also consider shape based attributes. These shape based attributes can either be indicators of global object shape (*e.g.*, “shoulder bag”) or indicators of local object shape (*e.g.*, “ankle strap”) depending on whether they refer to an entire object or part of an object. For each potential attribute we train a MILBoost classifier on three different feature types (color, texture, or shape – visual representation described in Section 4.2). The best performing feature measured by average precision is selected as the type.

## 4 Experimental Evaluation

We have performed experiments evaluating all aspects of our method: predicting visualness (Section 4.3), predicting the localizability (Section 4.4), and predicting type (Section 4.4). First we begin with a description of the data (Section 4.1), and the visual and textual representations (Section 4.2).

### 4.1 Data

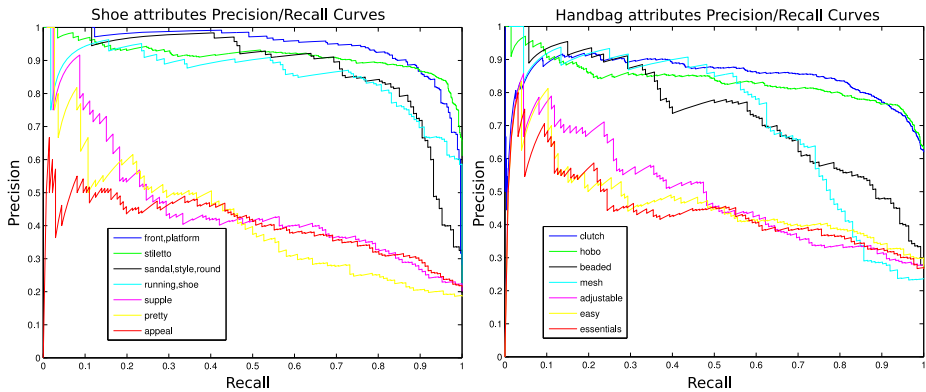
We have collected a large data set of product images from the internet<sup>1</sup> depicting four broad types of objects: shoes, handbags, earrings, and ties. In total we have 37795 images: 9145 images of handbags, 14765 images of shoes, 9235 images of earrings, and 4650 images of ties. Though these images were collected from a single website aggregator, they originate from a variety of over 150 web sources (*e.g.*, shoemall, zappos, shopbop), giving us a broad sampling of various categories for each object type both visually and textually (*e.g.*, the shoe images depict categories from flats, to heels, to clogs, to boots). These images tend to be relatively clean, allowing us to focus on the attribute discovery goal at hand without confounding visual challenges like clutter, occlusion etc.

On the text side however the situation is extremely noisy. Because this general web data, we have no guarantees that there will be a clear relationship between the images and associated textual description. First there will be a great number of associated words that are not related to visual properties (see fig 2). Secondly, images associated with an attribute phrase might not depict the attribute at all, also, and quite commonly, images of objects exhibiting an attribute might not contain the attribute phrase in their description (see fig 3). As the images originate from a variety of sources, the words used to describe the same visual attribute may vary. All these effects together produce a great deal of *noise in the labeling* that can confound the training of a visual classifier.

### 4.2 Representations

**Visual Representation:** We use three visual feature types: color, texture, and shape. For predicting the visualness of a proposed attribute we take a global descriptor approach and compute whole image descriptors for input to the AnyBoost framework. For predicting localizability and attribute type, we take a local

<sup>1</sup> Specifically from like.com, a shopping website that aggregates product data from a wide range of e-commerce sources.



**Fig. 7.** Quantitative Evaluation: Precision/Recall curves for some highly visual attributes, and other less visual attributes for shoes (left) and handbags (right). Each colored plot line shows precision/recall for a particular attribute. Some attributes (*e.g.*, stiletto) are more visual than others (*e.g.*, appeal).

descriptor approach, computing descriptors over local subwindows in the image with overlapping blocks sampled over the image (with block size 70x70 pixels, sampled every 25 pixels).

Each of our three feature types is encoded as a histogram (integrated over the whole image for global descriptors, or over individual regions for local descriptors), making selection and computation of decision stumps for our boosted classifiers easy and efficient. For the shape descriptor we utilize a SIFT visual word histogram. This is computed by first clustering a large set of SIFT descriptors using  $k$ -means (with  $k = 100$ ) to get a set of visual words. For each image, the SIFT descriptors are computed on a fixed grid across the image, then the resulting visual word histograms are computed. For our color descriptor we use a histogram computed in HSV with 5 bins for each dimension. Finally, for the texture descriptor we first convolve the image with a set of 16 oriented bar and spot filters [25], then accumulate absolute response to each of the filters in a texture histogram.

**Textual Representation:** On the text side we keep our representation very simple. After converting to lower case, removing stop words, and punctuation, we consider all remaining strings of up to 4 consecutive words that occur more than 200 times as potential attributes.

### 4.3 Evaluating Visualness Ranking

Some attribute synsets are shown for shoes in figure 5 and for handbags in figure 4, where each row shows some highly ranked images for an attribute synset. For shoes, the top 5 rows show highly ranked visual attributes from our collection: “front platform”, “sandal style round”, “running shoe”, “clogs”, and “high heel”. The bottom 3 rows show less highly ranked visual attributes: “great”,

Human Evaluation for Earring Attributes
<b>Human Based Classification</b>
Visual Attributes: "basket,setting", "solitaire,stud", "earring, studs,crafted", "heart", "screw,back", "princess","rating", "natural", "diamond,stud", "comes"
Non-Visual Attributes: "measure", "cz", "measures", "dangle", "quality", "anything,favorite, woman", "hoops", "outfit", "piece", "5mm"
<b>Our Classification</b>
Visual Attributes: "earring,studs,crafted", "screw,back", "rating", "solitaire,stud", "basket,setting", "anything,favorite,woman", "hoops", "princess","diamond,stud", "heart"
Non-Visual Attributes: "natural", "comes", "quality", "dangle", "5mm","piece", "cz", "outfit", "measure", "measures"

**Fig. 8.** Attributes from the top and bottom of our visualness ranking for earrings as compared to a human user based attribute classification. The user based attribute classification produces similar results to our automatic method (80% agreement for earrings, 70% for shoes, 80% for handbags, and 90% for ties).

“feminine”, and “appeal”. Note that the predicted visualness seems reasonable. This is evaluated quantitatively below. For handbags, attributes estimated to be highly visual include terms like “clutch”, “hobo”, “beaded”, “mesh” etc. Terms estimated by our system to be less visual include terms like “look”, “easy”, “adjustable” etc.

The visualness ranking is based on a **quantitative evaluation** of the classifiers for each putative attribute. Precision recall curves on our evaluation set for some attributes are shown in figure 7 (shoe attributes left, handbag attributes right). Precision and recall are measured according to how well we can predict the presence or absence of each attribute term in the images textual descriptions. This measure probes both the underlying visual coherence of an attribute term, and whether people tend to use the term to describe objects displaying the visual attribute. For many reasonable visual attributes our boosted classifier performs quite well, getting average precision values of 95% for “front platform”, 91% for stiletto, 88% for “sandal style round”, 86% for “running shoe” etc. For attributes that are probably less visual the average precision drops to 46% for “supple”, 41% for “pretty”, and 40% for “appeal”. This measure allows us to reasonably predict the visualness of potential attributes.

Lastly we obtain a **human evaluation** of visualness and compare the results to those produced by our automatic approach. For each broad category types, we evaluate the top 10 ranked visual attributes (classified as visual by our algorithm), and the bottom 10 ranked visual attributes (classified as non-visual by our algorithm). For each of these proposed attributes we show 10 labelers (using Amazon’s Mechanical Turk) a training set of randomly sampled images with that attribute term and without the term. They are then asked to label novel query images, and we rank the attributes according to how well their labels predict the presence or absence of the query term in the corresponding descriptions. The top half of this ranking is considered visual, and the bottom half as non-visual (see *e.g.*, fig 8). Classification agreement between the human method

and our method is: 70% for shoes, 80% for earrings, 80% for bags, and 90% for ties, demonstrating that our method agrees well with human judgments of attribute visualness.

#### 4.4 Evaluating Characterization

**Localizability:** Some examples of highly localizable attributes are shown in the top 4 rows of figure 6. These include attributes like “tote”, where MILBoost has selected the handle region of each bag as the visual representation, and “stiletto” which selects regions on the heel of the shoe. For “sandal style round” the open toe of the sandal is selected as the best attribute indicator. And, for “asics” the localization focuses on the logo region of the shoe which is present in most shoes of the asics brand. Some examples of more global attributes are shown in the bottom 2 rows of figure 6. As one might expect, some less localizable attributes are based on color (*e.g.*, “blue”, “red”) and texture (*e.g.*, “paisley”, “striped”).

**Type:** Attribute type categorization works quite well for color attributes, predicting “gold”, “white”, “black”, “silver” etc as colors reliably in each of our 4 broad object types. One surprising and interesting find is that “wedding” is labeled as a color attribute. The reason this occurs is that many wedding shoes use a similar color scheme that is learned as a good predictor by the classifier. Our method for predicting type also works quite well for shape based attributes, predicting “ankle strap”, “high heel”, “chandelier”, “heart”, etc to be shape attributes. Texture characterization produces more mixed results, characterizing attributes like “striped”, and “plaid” as texture attributes, but other attributes like “suede” or “snake” as SIFT attributes (perhaps an understandable confusion since both feature types are based on distributions of oriented edges).

## 5 Conclusions and Future Work

We have presented a method to automatically discover visual attributes from noisy web data. The method is able to reliably find and rank potential attribute phrases according to their visualness – a score related to how strongly a string is correlated with some aspect of an object’s visual appearance. We are further able to characterize attributes as localizable (referring to the appearance of some consistent subregion on the object) or global (referring to a global appearance aspect of the object). We also categorize attributes by type (color, texture, or shape). Future work includes improving the natural language side of the system to complement the vision-centric ideas presented here.

## References

1. Kumar, N., Berg, A.C., Belhumeur, P., Nayar, S.K.: Attribute and simile classifiers for face verification. In: ICCV (2009)
2. Kumar, N., Belhumeur, P., Nayar, S.K.: FaceTracer: A search engine for large collections of images with faces. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 340–353. Springer, Heidelberg (2008)



3. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
4. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.A.: Describing objects by their attributes. In: CVPR (2009)
5. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
6. Fei-Fei, L., Fergus, R., Perona, P.: A Bayesian approach to unsupervised one-shot learning of object categories. In: ICCV (2003)
7. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
8. Barnard, K., Duygulu, P., Forsyth, D.A.: Clustering art. In: CVPR (2005)
9. Yanai, K., Barnard, K.: Image region entropy: A measure of ‘visualness’ of web images associated with one concept. In: WWW (2005)
10. Barnard, K., Duygulu, P., de Freitas, N., Forsyth, D., Blei, D., Jordan, M.I.: Matching words and pictures. JMLR 3, 1107–1135 (2003)
11. Wang, J., Markert, K., Everingham, M.: Learning models for object recognition from natural language descriptions. In: BMVC (2009)
12. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS (2007)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI 99 (2009)
14. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly supervised object recognition and localization with stable segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 193–207. Springer, Heidelberg (2008)
15. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L., et al: Pascal Voc Workshops (2005-2009)
16. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: ICCV (2005)
17. Sivic, J., Russell, B.C., Zisserman, A., Freeman, W.T., Efros, A.A.: Discovering object categories in image collections. In: ICCV (2005)
18. Berg, T.L., Forsyth, D.A.: Animals on the web. In: CVPR (2006)
19. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. In: ICCV (2007)
20. Berg, T.L., Berg, A.C.: Finding iconic images. In: CVPR Internet Vision Workshop (2009)
21. Berg, T.L., Berg, A.C., Edwards, J., Forsyth, D.A.: Who’s in the picture? NIPS (2004)
22. Collins, B., Deng, J., Li, K., Fei-Fei, L.: Towards scalable dataset construction: An active learning approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 86–98. Springer, Heidelberg (2008)
23. Viola, P., Plattand, J.C., Zhang, C.: Multiple instance boosting for object detection. In: NIPS (2007)
24. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: CVPR (2005)
25. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. IJCV 43, 29–44 (2001)

# Learning to Recognize Objects from Unseen Modalities

C. Mario Christoudias<sup>1</sup>, Raquel Urtasun<sup>2</sup>,  
Mathieu Salzmann<sup>1</sup>, and Trevor Darrell<sup>1</sup>

<sup>1</sup>UC Berkeley EECS & ICSI

<sup>2</sup>TTI Chicago






**Abstract.** In this paper we investigate the problem of exploiting multiple sources of information for object recognition tasks when additional modalities that are not present in the labeled training set are available for inference. This scenario is common to many robotics sensing applications and is in contrast with the assumption made by existing approaches that require at least some labeled examples for each modality. To leverage the previously unseen features, we make use of the unlabeled data to learn a mapping from the existing modalities to the new ones. This allows us to predict the missing data for the labeled examples and exploit all modalities using multiple kernel learning. We demonstrate the effectiveness of our approach on several multi-modal tasks including object recognition from multi-resolution imagery, grayscale and color images, as well as images and text. Our approach outperforms multiple kernel learning on the original modalities, as well as nearest-neighbor and bootstrapping schemes.

## 1 Introduction

Recent advances in object recognition have shown that exploiting multiple sources of information could significantly improve recognition performance. This is the case when relying either on different image features [1–3], or on multiple modalities such as images and text [4–6]. Typically, the different inputs are combined either via kernels (i.e., multiple kernel learning) [1–3, 7] or by voting schemes [8, 9]. While these techniques have proven successful, they assume that all the modalities (features) are present during both training and inference.

However, this assumption is often violated in more dynamic scenarios where new modalities are added during inference. This is common in robotics applications, where a robot can be equipped with new sensors (e.g., high resolution cameras, laser range finders). Even though existing techniques can handle a certain degree of missing data, they all require some labeled examples for each modality. As a consequence, the only way for them to exploit these new modalities is to manually label some examples and re-train the classifier.

In this paper, we tackle the problem of exploiting novel modalities that are present only at test time for which no labeled samples are provided (see Fig. 1). This scenario is particularly challenging since the number of unlabeled examples might be small. To be able to leverage the previously unseen features, we assume that the conditional distribution of the new modalities given the existing ones is

	Conventional			Our approach			
Labeled training set							Low resolution
Unlabeled test set							Low resolution
							High resolution
Performance	75%	5%	33%	92%	29%	58%	

**Fig. 1. Exploiting unseen modalities:** In this paper we propose a new object recognition approach that can leverage additional modalities that are fully unlabeled. The example above illustrates how additional unlabeled high-resolution images let us significantly boost the classification performance over the low-resolution feature channel. Similar behavior is shown in our experiments when adding unlabeled color images to grayscale ones, and when using text in conjunction with images.

stationary. This is similar in spirit to the assumption typically made by semi-supervised learning techniques that the labeled and unlabeled examples are drawn from the same distribution. This lets us exploit the unlabeled data to learn a non-linear mapping from the existing modalities to the novel ones. From the resulting mapping, we “hallucinate” the missing data on the labeled training examples. This allows us to exploit the full potential of multiple kernel learning by using both old and new modalities.

As a result, our classifier improves over the original one by effectively making use of all the available modalities while avoiding the burden of manually labeling new examples. This is of crucial importance to make recognition systems practical in applications such as personal robotics, where we expect the users to update their robot, but cannot expect them to label a set of examples each time a new sensor is added.

We demonstrate the effectiveness our approach on a variety of real-world tasks: we exploit unlabeled high-resolution images to improve webcam object recognition, we utilize unlabeled color images for grayscale object recognition, we use unlabeled text to improve visual classification, and we exploit unlabeled images for sense disambiguation in the text domain. In all these scenarios we show that our method significantly outperforms multiple kernel learning on the labeled modalities, as well as nearest-neighbor and bootstrapping schemes.

## 2 Related Work

Many techniques have been proposed that exploit multiple feature cues or information sources for performing object recognition. A popular approach is to

use multiple kernel learning (MKL) either in an SVM framework [1, 3] or in a Gaussian processes probabilistic framework [2]. Voting schemes have also been proposed for multi-feature object recognition [8, 9]. In [9] the implicit shape model (ISM) was extended to include multiple features, while in [8] a naive-Bayes nearest-neighbor classifier within a voting-based scheme was utilized. These multiple feature recognition approaches have been shown to be highly beneficial and lead to state-of-the-art performance on several challenging problems [7]. However, these approaches have focused on supervised or semi-supervised scenarios where at least some labels are provided for each modality, and cannot exploit additional unsupervised modalities available only at test time.

Semi-supervised multi-view learning approaches have been used to exploit both labeled and unlabeled data. In [10] co-training was used to learn an object detector. Bayesian co-training was explored in [11] for instance-level object recognition. Similarly, multi-view bootstrapping schemes were used in [12] to transcribed speech and video features for video concept detection, and in [13] to learn audio-visual speech and gesture classifiers. Still, most of these approaches make the assumption that at least some labels are provided for each modality. The exception being cross-modal bootstrapping [13] that can leverage a classifier from a single view to learn a multi-view classifier. However, as demonstrated in our experiments this approach does not take full advantage of the unlabeled modalities.

Methods for learning a joint latent space from multiple modalities have also been proposed. In [4, 5] latent Dirichlet allocation (LDA) was used to perform visual sense disambiguation using unsupervised text and images. In [14] a transfer learning approach was proposed to learn a discriminatively trained latent space over images and their captions. Such methods can be seen as complementary to our approach in that we also exploit a form of information transfer between modalities to infer the missing ones. Yet, to our knowledge, no previous approach has considered the problem of having modalities for which no labeled examples are provided, and this is the first attempt to do so.

The most related work to ours is probably [6] where they employ a nearest-neighbor approach to infer text histograms from images using a large external collection of images and text captured from the web. This is different from our approach in that their method is specifically designed to infer text and assumes that a very large dataset (i.e., hundreds of thousands of examples) is available. As evidenced by our experiments, our approach significantly improves over nearest-neighbor inference across a wide range of problems.

### 3 Exploiting Unseen Modalities

In this section, we present our approach to exploiting new modalities at test time even though there is no labeled data for them. Towards this end, we show how to hallucinate the missing modalities for the labeled examples by learning a mapping from the old modalities to the new ones from the unlabeled data. Given these hallucinated modalities, we propose a framework that combines the different sources of information using probabilistic multiple kernel learning. We

then introduce a representation of the novel modalities that lets us exploit the full potential of non-linear kernels recently developed for object recognition, while still making regression possible. Finally, we present a bootstrapping algorithm that further improves the performance of our classifier.

### 3.1 “Hallucinating” the Missing Modalities

To leverage the availability of fully unsupervised modalities for classification, we propose to infer these missing modalities for the labeled examples and use them in conjunction with the old modalities in a probabilistic multiple kernel learning framework. In this section we show how to hallucinate the missing modalities.

More formally, let  $\mathbf{X} = [\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(D)}]$  be the set of training inputs for the  $D$  modalities present in both the labeled and unlabeled datasets, with  $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \dots, \mathbf{x}_{N_{train}}^{(i)}]^T$  the  $N_{train}$  training points for the  $i$ -th modality, and let  $\bar{\mathbf{X}} = [\bar{\mathbf{X}}^{(1)}, \dots, \bar{\mathbf{X}}^{(D)}]$  be the test examples, with  $\bar{\mathbf{X}}^{(i)} = [\bar{\mathbf{x}}_1^{(i)}, \dots, \bar{\mathbf{x}}_{N_{test}}^{(i)}]^T$ . Let  $\bar{\mathbf{Z}} = [\bar{\mathbf{Z}}^{(1)}, \dots, \bar{\mathbf{Z}}^{(M)}]$  be the  $M$  new modalities present only in the unlabeled test set, such that  $\bar{\mathbf{Z}}^{(i)} = [\bar{\mathbf{z}}_1^{(i)}, \dots, \bar{\mathbf{z}}_{N_{test}}^{(i)}]^T$ .

In order to exploit the new modalities that are only present at test time and for which we do not have any labeled examples, we assume that the conditional distribution of the new modalities given the labeled ones is stationary, i.e., the same at training and inference. This is similar in spirit to the standard assumptions of semi-supervised learning techniques, and lets us rely on the concept of mean imputation typically used when dealing with missing data. However, here we assume that only a small amount of unlabeled data is available to learn the mapping from the known modalities  $\mathbf{x}$  to the new modalities  $\mathbf{z}$ . This makes the problem more challenging, since simple methods such as nearest-neighbors (NN) require large collections of examples for accurate prediction.

To overcome this issue, we rely on Gaussian processes (GPs) which have proven effective when trained from a small number of examples [15]. This is due to the fact that they marginalize among all possible non-linear mappings defined by the kernel function. In particular, we utilize a GP to learn the mapping from the known modalities to the missing ones. Note that unlike for the classification task, when hallucinating the new modalities the unlabeled examples are used as training data, since for those both  $\bar{\mathbf{Z}}$  and  $\bar{\mathbf{X}}$  are known. Under this model, the likelihood can be expressed as

$$p(\bar{\mathbf{Z}}|\bar{\mathbf{X}}) = \prod_{m=1}^M \prod_{i=1}^{S_m} p(\bar{\mathbf{z}}_{:,i}^{(m)}|\bar{\mathbf{X}}) = \prod_{m=1}^M \prod_{i=1}^{S_m} \mathcal{N}(\bar{\mathbf{z}}_{:,i}^{(m)}; 0, \mathbf{K}^x), \tag{1}$$

where  $S_m$  is the dimensionality of the  $m$ -th new modality. The elements of the kernel associated with the labeled modalities  $\mathbf{K}^x$  are computed by kernel combination as

$$\mathbf{K}_{i,j}^x = \sum_{m=1}^D \alpha_m k^x(\bar{\mathbf{x}}_i^{(m)}, \bar{\mathbf{x}}_j^{(m)}), \tag{2}$$

where  $\alpha_m$  are hyper-parameters of the model. In order to capture the correlations between the different output dimensions and modalities, we share kernel hyper-parameters across the different predictors.

Given the known modalities  $\mathbf{x}_i$  for a labeled example, the predictive distribution under the Gaussian process is also Gaussian and can be computed in closed form, i.e.,  $p(\mathbf{z}_i|\mathbf{x}_i, \bar{\mathbf{X}}, \bar{\mathbf{Z}}) = \mathcal{N}(\mu(\mathbf{x}_i), \sigma(\mathbf{x}_i))$ , with mean and variance

$$\mu(\mathbf{x}_i) = \mathbf{k}_i^x (\mathbf{K}^x)^{-1} \bar{\mathbf{Z}} \quad (3)$$

$$\sigma(\mathbf{x}_i) = k^x(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{k}_i^x (\mathbf{K}^x)^{-1} \mathbf{k}_i^{xT}, \quad (4)$$

where  $\mathbf{k}_i^x$  is the vector obtained by evaluating the kernel function of Eq. (2) between  $\bar{\mathbf{X}}$  and  $\mathbf{x}_i$ . For each labeled example, we hallucinate the missing modalities by taking them as the mean prediction of the learned GP. The resulting hallucinated modalities predicted by the GP can then be used in conjunction with the labeled ones in the probabilistic multiple kernel learning framework described in the following section. Note that here we have made the assumption that the mapping between the old and new modalities is unimodal, i.e., can be modeled with a GP. As suggested by our results, this assumption is reasonable for a wide range of problems. In more challenging scenarios, it can easily be relaxed by using a mixture of local predictors [16].

### 3.2 Probabilistic Multiple Kernel Learning

To exploit all available sources of information for classification, we combine the hallucinated modalities and the old ones within a probabilistic multiple kernel learning framework. In particular, we employ GPs to learn the mapping from  $(\mathbf{x}, \mathbf{z})$  to the labels  $y$ . This has been shown to perform similarly to SVM-based MKL approaches while being computationally more efficient [2]. This yields the likelihood  $p(\mathbf{y}|\mathbf{X}, \mathbf{Z}) = \mathcal{N}(\mathbf{y}; 0, \mathbf{K})$ , where  $\mathbf{y} = [y_1, \dots, y_N]^T$  are the labels for the training examples and the elements of  $\mathbf{K}$  are computed as

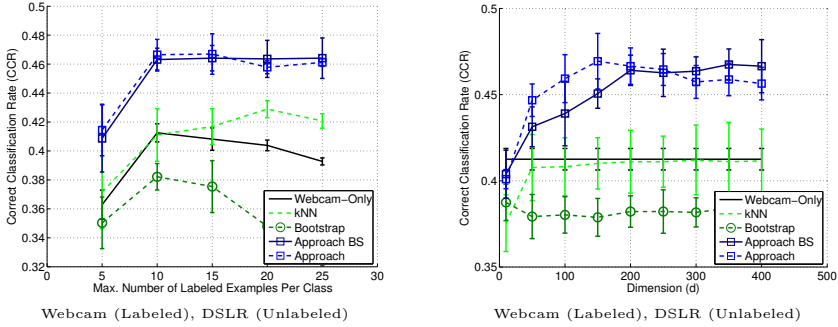
$$\mathbf{K}_{i,j} = \mathbf{K}_{i,j}^x + \sum_{m=1}^M \beta_m k^z(\mathbf{z}_i^{(m)}, \mathbf{z}_j^{(m)}) \quad , \quad (5)$$

with  $k^z(\cdot, \cdot)$  the kernel function for the unsupervised modalities.

Given new input observations  $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ , we use the mean prediction of the GP to assign a class label to the example. For multi-class problems we used a one-vs-all strategy that selects the class having the largest positive mean prediction. Note that we use a Gaussian noise model which has been shown to perform similarly to more complex noise models (e.g., probit, logit) [17].

### 3.3 A General Representation of the Novel Modalities

While for some representations of  $\mathbf{z}$  (e.g., histograms) the above framework could be effective, we would like our MKL algorithm to be able to exploit complex non-linear kernels (e.g., Pyramid Match Kernel (PMK) [18], Spatial Pyramid [19]) that have proven successful for object recognition tasks. Note that, with these



**Fig. 2. Robotics dataset:** We consider the scenario where only low-resolution webcam images are labeled and an additional unlabeled high-resolution modality is available at test time using the dataset of [21]. (left) Comparison of our approach against several baselines as a function of the number of labeled examples, Approach BS is our approach with bootstrapping. (right) Accuracy as a function of K-PCA dimensionality for  $Q = 10$ . Note in both cases our approach outperforms the baselines. Error bars indicate  $\pm 1$  std.

kernels,  $\mathbf{z}$  would correspond to the high-dimensional feature map which cannot be computed in practice.

We overcome this difficulty by learning a representation of the unlabeled modalities that is able to exploit complex non-linear kernels. To this end, we rely on Kernel PCA [20], which computes a low-dimensional representation of the possibly infinite dimensional input space by performing the eigendecomposition of the (centered) kernel matrix evaluated on the unlabeled data. This representation is interesting since it is low-dimensional and allows us to use any Mercer kernel to represent the new modalities. Our approach then proceeds as follows: First, we compute K-PCA on the new modalities and retain the first  $d$  dimensions. We then regress from the old modalities to the K-PCA representation of the new ones to hallucinate the missing data. Given the hallucinated data, we perform probabilistic multiple kernel learning with all the modalities, using a kernel computed in the K-PCA embedding for the new modalities.

### 3.4 Bootstrapping

To make further use of the unlabeled examples, we propose to use a bootstrapping strategy. At first, our multiple kernel classifier is trained on all the modalities using the hallucinated data. We then evaluate it on the unlabeled data and add the  $B$  most confident predictions per class to the set of labeled examples. For the confidence measure, we rely on the distance from the mean prediction to the predicted class label. This is similar to the concept of margin in SVMs. Note that other criteria used for active learning could be employed, e.g., uncertainty [2]. Given the old and new labeled examples, we train a new classifier and repeat the process  $T$  times.

## 4 Experimental Evaluation

We evaluated our approach on a broad range of object recognition domains where modalities that are available at test time may not be present in the labeled training set. In particular, we first show how we can classify high-resolution images when only low-resolution webcam images are labeled. We then show how exploiting unlabeled color images in conjunction with intensity-only labeled data improves classification performance. Finally, we address the problem of classification of text and image datasets, where only either text or images are labeled. Across all of these real-world problem domains, our approach achieves a significant performance boost by exploiting the additional test modalities without requiring any new labeled samples. The paper code and databases are available online at <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/msorec/>.

**Baselines:** On each dataset we compare against two baselines. The first one employs  $k$  nearest-neighbor ( $k$ -NN) to infer the missing modalities by averaging across the  $k$  neighbors. This approach is akin to [6] where they rely on  $k$ -NN to infer text features from a large collection of web images. On each dataset we chose the  $k$  that gave the best performance. Note that the baseline makes use of the test data to select the best parameter  $k$ , while our approach does not. The second baseline (Bootstrap) exploits the additional test modalities by cross-modal bootstrapping analogous to [13]; an initial seed set is formed by labeling  $B_{init}$  examples per class using the single-view classifier, and the same bootstrapping strategy used by our approach is then applied. We also show single-view and oracle performance for each dataset, where the oracle makes use of the ground-truth features on the missing modalities for the labeled training set.

**Experimental setup:** For the bootstrapping baseline and our approach, we used  $B = 2$ , and  $T = 10$  across all datasets, and  $B_{init} = 10$  for the baseline. We used RBF kernels whose bandwidths were set to the mean squared distance over the dataset to compute K-PCA and for GP regression. Additionally, we set the Gaussian noise variance to 0.01. For all but the Mouse dataset, we used  $d = 200$ . For this dataset we used  $d = 20$ , since the number of examples is much smaller than in the other datasets. In the case of multiple kernel classifiers, we set  $\alpha_m$  and  $\beta_m$  to be  $1/V$  where  $V$  is the total number of views. We did not optimize  $\alpha_m$  and  $\beta_m$ , since it has been shown that averaging the kernels yields similar performance [2, 22]. We ran each experiment on 5 random splits of the data into training and test sets where for each split we used  $Q$  labeled examples per class. The performance of each approach was evaluated using the Correct Classification Rate (CCR) defined as the total number of correctly classified examples divided by the total number of examples.

### 4.1 Multi-sensor Object Recognition

We first consider a multiple sensor robotics scenario and show that our approach lets us leverage the better quality of high resolution images even though only low resolution images were labeled. We used a subset of the images from the dataset

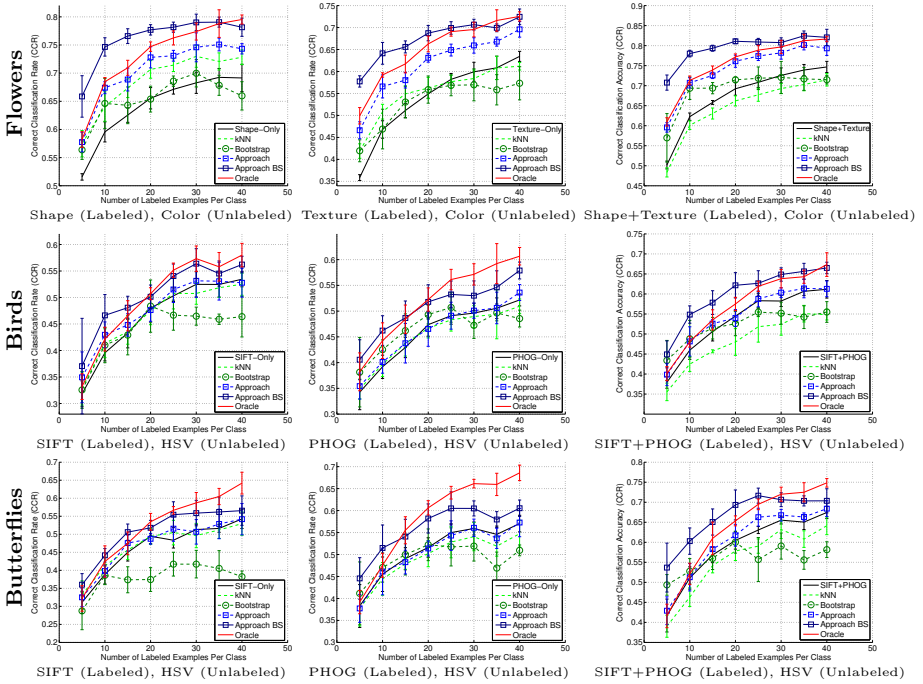




**Fig. 3. Most confident images in the robotics application:** For each class, the top row shows the most confident images returned by the low-resolution only classifier, and the bottom row depicts similar images for our classifier trained with additional unlabeled high-resolution images. Our approach significantly improves the results for some of the classes and has less impact on others. Nonetheless, even for these classes it reduces the ambiguity, e.g., to only two different labels.

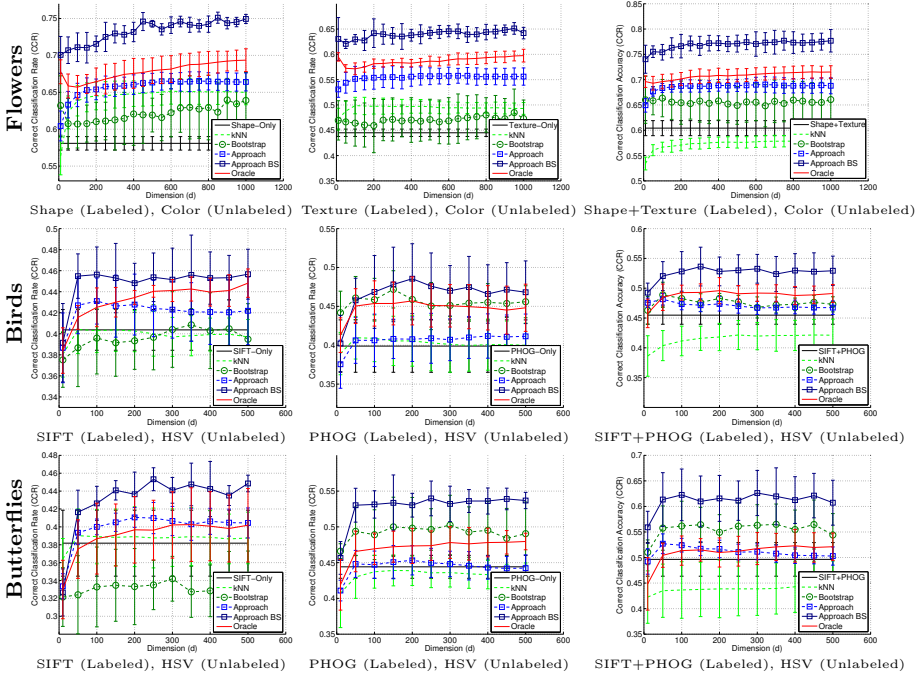
of [21] yielding 783 low resolution and 486 high resolution images of 30 office object categories captured using a webcam and a DSLR camera. We split the dataset into webcam-only images and webcam+DSLR image pairs that depict an object from similar viewpoints. This resulted in a total of 368 webcam images used as labeled examples and 415 webcam+DSLR image pairs that we treated as our unlabeled set. For each sensor type, we used PHOG features [23] with 4 pyramid levels and 8 histogram bins per cell over 360 degrees. We then applied PCA to those vectors and retained 95% of the variance to form the final feature vector.

This dataset is fairly challenging as it consists of a wide variety of object categories and appearances taken from a sparse set of varying viewpoints. Fig. 2



**Fig. 4. Using unlabeled color images for grayscale object recognition:** For each dataset, we show (left,middle) the performance of our approach using only a single intensity feature, and (right) using both of the available intensity features. Note that our approach achieves a significant performance boost over intensity-only performance and outperforms the other baselines. Performance is shown across different training sizes with  $d = 200$ . The error bars indicate  $\pm 1$  std.

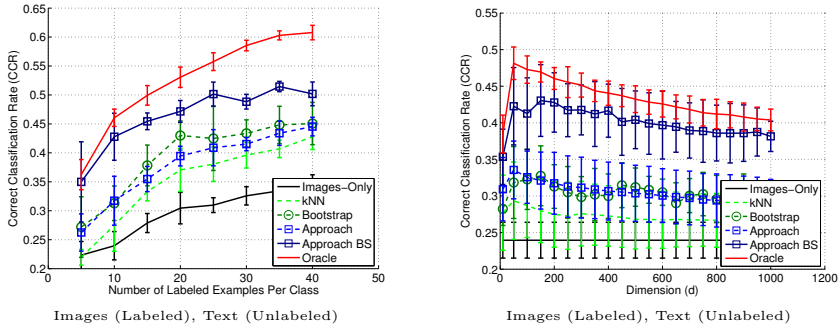
(left) depicts performance as a function of the training set size. For this dataset, at most  $Q$  labeled examples were retained per class as some classes had fewer than  $Q$  webcam images across the different training set sizes. Since the number of unlabeled examples is fairly small in this dataset, the performance of the  $k$ -NN baseline is poor. For similar reasons, the cross-modality bootstrap baseline is unable to improve over the weak performance of the webcam-only classifier. In contrast, our approach that infers the missing DSLR modality on the labeled training set results in a stronger multi-view classifier. Fig. 2 (right) depicts performance with varying K-PCA dimensionality for  $Q = 10$  training samples per class. Our approach proves to be fairly insensitive to the choice of  $d$  and outperforms the baselines across a wide range of dimensionalities of the embedding. In Fig. 3, we compare the most confident images returned by the webcam-only classifier and by our approach. As can be observed from the images, our classifier is able to avoid some of the mistakes of the webcam-only baseline.



**Fig. 5. Using unlabeled color images for grayscale object recognition:** Classification error as a function of the K-PCA dimensionality for  $Q = 10$ . The plots show the performance using a single labeled intensity feature (left,middle) and using both available intensity features (right) to classify the color images. Our approach is insensitive to the choice of dimensionality and improves over the baselines.

### 4.2 Using Unlabeled Color Images for Grayscale Object Recognition

We now illustrate how our approach is capable of exploiting the additional information contained in unlabeled color images to improve the performance of a grayscale object classifier, when only grayscale examples are labeled. This is a plausible scenario in robotics applications where robots are equipped with high-performance (e.g., hyperspectral) cameras. For this task we used three datasets of natural object categories, where color features are relevant for classification. The first dataset is the Oxford Flowers dataset [3] that is comprised of 80 images of 17 flower categories. For this dataset, the authors have provided  $\chi^2$  distance matrices over three visual feature channels: Color, Shape and Texture. The other two datasets are comprised of butterfly and bird image categories respectively [24, 25]. The Birds dataset contains 6 classes with 100 images per class and the Butterflies dataset has 619 images of 7 classes. For these datasets we extracted dense SIFT [26], PHOG [23] and HSV color features [3]. Images were compared using  $\chi^2$  distances for SIFT and HSV features and  $L_2$  distance for PHOG.



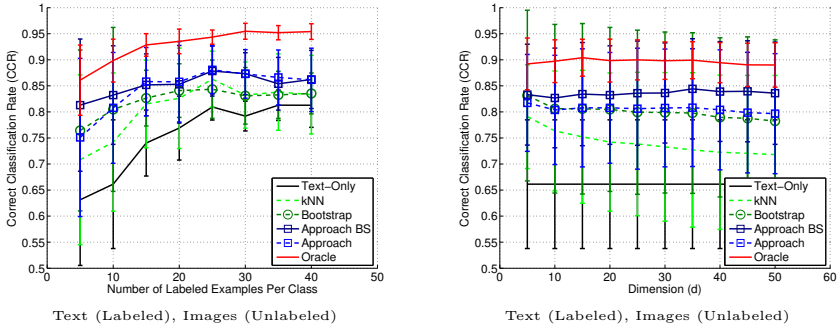
**Fig. 6. Using unlabeled text to improve visual classification:** We focus on a recognition task that exploits an additional text modality that is only present at test time (no labeled data is available) to improve the performance of a visual classifier. We used the dataset of [5] that consists of a set of noisily labeled images obtained from an online keyword search. Using unlabeled text features our approach is able to significantly improve performance over the weakly supervised image classifier. Performance is shown across different training set sizes with  $d = 200$  (left) and across different K-PCA dimensions with  $Q = 10$  (right). Error bars indicate  $\pm 1$  std.

Fig. 4 displays performance on the three datasets as a function of the training set size. Note that unlike  $k$ -NN, our approach is able to obtain a good estimate of the missing color modality from few training samples, and significantly improves over grayscale-only performance. The conventional bootstrapping baseline is also unable to achieve a significant improvement and often underperforms grayscale-only performance. In contrast, an even greater improvement is achieved with our approach when used in combination with bootstrap (Approach BS) that often matches or even improves upon the supervised oracle. Fig. 5 depicts performance for the three datasets as a function of the dimensionality of the K-PCA embedding for  $Q = 10$ . Note that the performance of our approach is relatively insensitive to the choice of dimensionality.

For completeness of our evaluation, we compared our approach and the baselines over all the possible feature combinations, even though these combinations do not necessarily represent real-world scenarios, e.g., having HSV and PHOG at test time but only labeled SIFT. Similar performance as to that of the above cases was observed. These plots are available at the project webpage listed above.

### 4.3 Using Unlabeled Text to Improve Visual Classification

Next, we focused on the task of leveraging unlabeled text features to improve over image-only object categorization. Our labeled set consists of a set of images collected using keyword search from an image search engine. Such a dataset can be considered as weakly supervised in that many images returned for a given object keyword can be only very loosely related to the target category. In the test set each image is accompanied with text (e.g., extracted from the webpage). We used the Office dataset of [5] that consists of text and images for 10 object



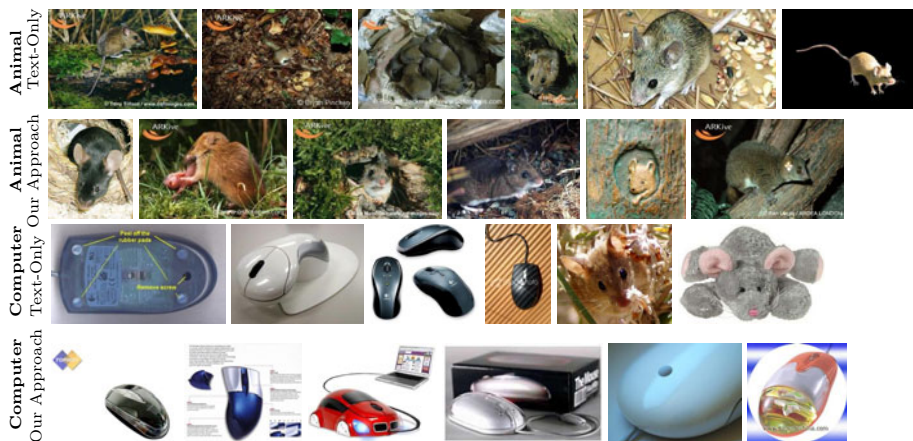
**Fig. 7. Using unlabeled images for sense disambiguation in text classification:** We tackle the problem of sense disambiguation from a labeled set of text examples and a set of unlabeled images that are available only at test time. We used the dataset of [4] to disambiguate the two meanings of the word mouse that pertain to the animal and the computer device classes. Our approach is able to significantly improve over a text-only classifier,  $k$ -NN and bootstrap baselines using only unlabeled images. Performance is shown across different training set sizes with  $d = 20$  (right) and across different dimensions with  $Q = 10$  (left). Error bars indicate  $\pm 1$  std.

categories. As this dataset is fairly large, we considered a subset of the data and randomly chose 200 examples per class to form our evaluation set. We used the same features as [5], which consists of histograms of SIFT features and word histograms. The histograms were compared using  $\chi^2$  distances.

Fig. [6] depicts performance as a function of the training set size for  $d = 200$ , and as a function of the dimensionality of the K-PCA embedding for  $Q = 10$ . As expected, due to the large amount of noise, performance using images alone is fairly weak. In contrast, our approach is able to leverage the additional unlabeled text modality and significantly improves recognition performance over the weakly supervised image classifier. When combined with bootstrapping, it nearly matches oracle performance. Both the  $k$ -NN and the cross-modality bootstrap baselines are also able to improve over image-only performance, although they do not perform as well as our method. Similarly as before, we can see that our approach is rather insensitive to the choice of  $d$ .

#### 4.4 Using Unlabeled Images for Sense Disambiguation in Text

We now consider the problem of exploiting unlabeled images to disambiguate the sense of classes described by labeled text features only. Sense disambiguation is important when each object category can pertain to multiple visual senses [4, 5]. For example, the keyword MOUSE can pertain to the animal or the computer device. Our goal is to use unsupervised images to improve the performance of a text-only classifier to discriminate polysemous object categories. We used a subset of the dataset of [4] that consists of about 100 examples per class, selected to contain images only of the target senses. Each image is represented using a histogram of dense SIFT features and each text document is summarized into a word histogram. Histograms are compared using the  $\chi^2$  distance.



**Fig. 8. Most confident images in sense disambiguation:** As in Fig. 3, the top row of each class shows the most confident images returned by the classifier built from labeled text features, and the bottom row depicts the result of our classifier when using images as an additional unsupervised modality. Note that, for the animal meaning, both approaches perform similarly whereas our classifier outperforms the text-only one on the computer sense.

Fig. 7 depicts performance as a function of the training set size for  $d = 200$  and of the dimensionality of the embedding for  $Q = 10$ . Although performance with text-only features is fairly good, the addition of the unlabeled visual modality significantly improves performance. Once again our approach outperforms the  $k$ -NN and cross-modality bootstrap baselines. Fig. 8 depicts some of the most confident images obtained for each class by either the text-only classifier, or by our approach. Note that our approach is able to avoid some of the mistakes made by the text-only classifier.

## 5 Conclusion

In this paper we have investigated the problem of exploiting multiple sources of information when some of the modalities do not have any labeled examples, i.e., are only present at test time. Assuming that the conditional distribution of novel modalities given old ones is stationary, we have shown how to make use of the unlabeled data to learn a non-linear mapping that hallucinates the new modalities for the labeled examples. Furthermore, we have shown how to learn low-dimensional representations that allow us to exploit complex non-linear kernels developed for object recognition. Finally, our approach is able to employ multiple kernel learning with all the modalities as well as a bootstrapping strategy that further improved performance. We have demonstrated the effectiveness of our approach on several tasks including object recognition from intensity and

color cues, text and images, and multi-resolution imagery. In the future we plan to investigate complementary techniques for inferring the missing views that include learning a shared latent space and the use of local GPs to cope with multi-modal output spaces.

## References

1. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV (2007)
2. Kapoor, A., Graumann, K., Urtasun, R., Darrell, T.: Gaussian processes for object categorization. IJCV (2009)
3. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: CVPR (2006)
4. Saenko, K., Darrell, T.: Unsupervised learning of visual sense models for polysemous words. In: NIPS (2008)
5. Saenko, K., Darrell, T.: Filtering abstract senses from image search results. In: NIPS (2009)
6. Wang, G., Hoiem, D., Forsyth, D.: Building text features for object image classification. In: CVPR (2009)
7. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: CVPR (2009)
8. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
9. Leibe, B., Mikolajczyk, K., Schiele, B.: Segmentation based multi-cue integration for object detection. In: BMVC (2006)
10. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual detectors using co-training. In: ICCV (2003)
11. Christoudias, C.M., Urtasun, R., Kapoor, A., Darrell, T.: Co-training with noisy perceptual observations. In: CVPR (2009)
12. Yan, R., Naphade, M.: Semi-supervised cross feature learning for semantic concept detection in videos. In: CVPR (2005)
13. Christoudias, C.M., Saenko, K., Morency, L.P., Darrell, T.: Co-adaptation of audio-visual speech and gesture classifiers. In: ICMI (2006)
14. Quattoni, A., Collins, M., Darrell, T.: Learning visual representations using images with captions. In: CVPR (2007)
15. Urtasun, R., Fleet, D., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: ICCV (2005)
16. Urtasun, R., Darrell, T.: Sparse probabilistic regression for activity-independent human pose inference. In: CVPR (2008)
17. Urtasun, R., Quattoni, A., Lawrence, N., Darrell, T.: Transferring nonlinear representations using gaussian processes with a shared latent space. Technical report, MIT (2008)
18. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV (2005)
19. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)

20. Scholkopf, B., Smola, A., Muller, K.: Kernel principal component analysis. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327. Springer, Heidelberg (1997)
21. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV (2010)
22. Gehlera, P., Nowozin, S.: Learning image similarity from flickr groups using stochastic intersection kernel machines. In: ICCV (2009)
23. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forest and ferns. In: ICCV (2007)
24. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: BMVC (2004)
25. Lazebnik, S., Schmid, C., Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In: CVPR (2005)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)



# Building Compact Local Pairwise Codebook with Joint Feature Space Clustering

Nobuyuki Morioka<sup>1,3</sup> and Shin'ichi Satoh<sup>2</sup>

<sup>1</sup> The University of New South Wales, Australia  
nmorioka@cse.unsw.edu.au

<sup>2</sup> National Institute of Informatics, Japan  
sato@nii.ac.jp

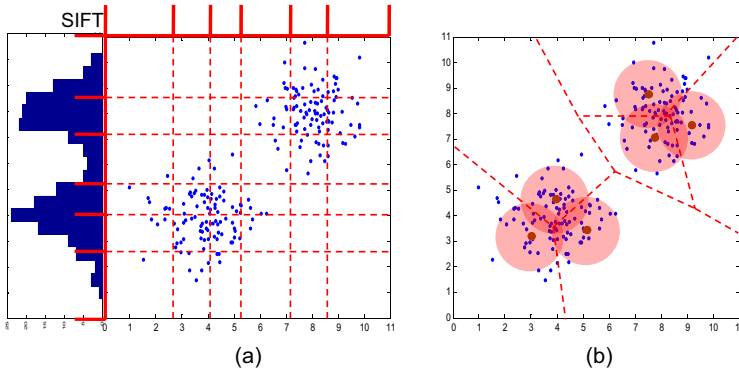
<sup>3</sup> NICTA, Australia

**Abstract.** This paper presents a simple, yet effective method of building a codebook for pairs of spatially close SIFT descriptors. Integrating such codebook into the popular bag-of-words model encodes local spatial information which otherwise cannot be represented with just individual SIFT descriptors. Many previous pairing techniques first quantize the descriptors to learn a set of visual words before they are actually paired. Our approach contrasts with theirs in that each pair of spatially close descriptors is represented as a data point in a joint feature space first and then clustering is applied to build a codebook called Local Pairwise Codebook (LPC). It is advantageous over the previous approaches in that feature selection over quadratic number of possible pairs of visual words is not required and feature aggregation is implicitly performed to achieve a compact codebook. This is all done in an unsupervised manner. Experimental results on challenging datasets, namely 15 Scenes, 67 Indoors, Caltech-101, Caltech-256 and MSRCv2 demonstrate that LPC outperforms the baselines and performs competitively against the state-of-the-art techniques in scene and object categorization tasks where a large number of categories need to be recognized.

**Keywords:** bag-of-words, spatial pyramid matching, higher-order spatial features, local pairwise codebook.

## 1 Introduction

For scene and object categorization tasks, the bag-of-words model has received much attention due to its simplicity and robustness. However, because of its orderless representation of local features, there have been numerous works [7, 8, 10, 11, 15, 17, 18, 25] that consider spatial arrangement of the features to discover higher-order structures inherent in scenes and objects for improved performance. Amongst the many, one simple method is to consider pairs of spatially close visual words [8, 10, 11]. This is commonly done by first obtaining a codebook where each feature descriptor is mapped to a visual word and then representing occurrences of local pairs of visual words with the bag-of-words model.



**Fig. 1.** (a) Traditional pairwise approach: The distribution of SIFT descriptors is viewed as one dimensional data and intervals shown in red indicate the derived clusters. The clusters are then used to determine the partition of the local pairwise distribution which is depicted in the dotted red line. (b) Our pairwise approach: The underlying local pairwise distribution is partitioned directly by joint feature space clustering to achieve a compact local pairwise codebook.

However, we speculate several issues with building the codebook prior to the pairing process. *First*, the number of possible pairs of visual words grows in quadratic with respect to the codebook size. This is depicted in Fig. 1 (a) where high dimensional descriptors like SIFT [12] are seen as one dimensional data and plotted along the x and y axes. Such quadratic growth results in a high dimensional histogram representation where in the case of a few training samples available, a classifier may over-fit and fail to generalize over testing data. *Second*, while feature selection [8,11] can be applied on these pairs of visual words to avoid the over-fitting problem, it often requires additional information like class labels and does not always lead to performance improvement [8]. *Third*, as illustrated in Fig. 1 (a), the traditional pairing approaches ignore the underlying distribution of pairs of nearby local feature descriptors, as they have already determined the partition of such distribution from the single feature codebook. We suspect that this way of over-partitioning the distribution may lead to poor generalization - degrading the recognition as previously seen in [8].

Given the above-mentioned issues, this paper describes a different approach, rather a reversed approach, in discovering a compact set of local pairwise feature clusters called *Local Pairwise Codebook* (LPC). We first concatenate each pair of spatially close descriptors and treat it as a data point in a joint feature space. Then, clustering is applied on the data to build a codebook as depicted in Fig. 1 (b). Our approach considers the underlying distribution explicitly, thus the partitioning is data-driven. In contrast to the traditional approaches where the number of pairwise feature clusters is determined by the single feature codebook, the size of LPC is controlled directly by a clustering algorithm and can be set to a moderate size. Thus, feature selection is not required to achieve a compact

codebook which is significantly smaller than quadratic number of possible pairs of visual words formed from a moderate-sized single feature codebook. With LPC, we directly encode local structure information into the bag-of-words model to boost the recognition in categorization tasks.

In summary, there are three main contributions in this paper. *Firstly*, we perform joint feature space clustering to build a compact local pairwise codebook based on SIFT descriptors to overcome issues discussed above. *Secondly*, when assigning the nearest cluster to each pair of nearby descriptors in an image, its time complexity grows in the number of pairs formed. Thus, to significantly reduce the complexity, we propose an efficient pairwise cluster assignment technique. *Thirdly*, we combine LPC with spatial pyramid matching kernel [9] to demonstrate that local and global spatial information complement each other to achieve competitive results on two scenes and three object categorization datasets, namely 15 Scenes [9], 67 Indoors [16], Caltech-101 [3], Caltech-256 [6] and MSRCv2 [22] datasets.

This paper is organized as follows. In Sect. 2, we cover some of the related work. Then Sect. 3 presents how to learn and use LPC in detail. This is followed by the experimental results in Sect. 4 to demonstrate the effectiveness of LPC. Finally, Sect. 5 concludes our paper with possible future work.

## 2 Related Work

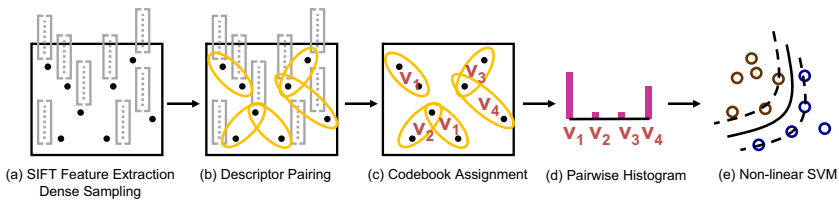
The idea of simply pairing up local features within their spatial local neighborhood is not new. While Lazebnik et. al. [8] have attempted to categorize objects by pairing visual words of SIFT descriptors in their maximum entropy framework, their results have shown that the pairwise features with frequency-based feature selection do not necessarily improve the performance over just using the visual words. Similar results are obtained by Lan et. al. [7]. Interestingly, both [7] and [8] have commented that higher-order spatial features might be more useful in locating and segmenting out objects of interest.

In contrast, Liu et. al. [11] have proposed an efficient feature selection method based on boosting which progressively mines higher-order spatial features. This has resulted in performance improvement up to  $2^{nd}$  order features, i.e. pairs of visual words. Savarese et. al. [17] have presented correlaton which captures the distribution of the distances between pairs of visual words based on clustering a set of correlogram elements. When this is combined with the appearance based bag-of-words model, it achieves promising results on some challenging object class recognition datasets. Instead of using actual spatial distance between local features, Proximity Distribution Kernel [10] and its variant [21] have reduced such information into ranks to achieve scale-invariance in their representation. Quak et. al. [15] have used frequent itemset mining to discover a set of distinctive spatial configurations of visual words to learn different object categories. In addition, such higher-order spatial features have shown to be also effective in unsupervised object discovery [18] and image retrieval [219].

All the above-mentioned techniques thus far assume that a set of visual words is already learned before considering pairs of features to represent higher-order

spatial information - indicating the issues discussed in the previous section. Our work is different from the above techniques in that we apply quantization in the joint feature space of local pairs of descriptors. A compact codebook of such pairs can be built by discovering clusters that encode correlation between spatially close descriptors. We believe that this captures better generalization of local pairwise feature distribution. Furthermore, it is possible to combine our work with the above techniques such as modeling the distribution of the distances between descriptors. From one perspective, our work can be seen as an extension of spatial pyramid matching kernel proposed by Lazebnik et. al. [9], as both approaches are based on densely sampled feature descriptors, but we consider local pairs of such descriptors to learn a more discriminative codebook.

### 3 Local Pairwise Codebook



**Fig. 2.** An overview of our approach. (a) Features are densely sampled and described by SIFT. (b) Features are paired locally and descriptors are concatenated. (c) A codeword is assigned to each local pair of features. (d) A histogram of pairwise codewords are established per image. (e) Non-linear SVM is used for image classification.

#### 3.1 Baseline: Pairing Visual Words

Before describing our approach, we first outline how pairs of visual words are usually formed in the previous approaches. Note that we only model occurrences of pairs of visual words within predefined neighborhood denoted as  $\gamma$ . In other words, other spatial information such as direction and distance between the pairs is not explicitly modeled. Each key-point  $f_i$  in an image is encoded as  $(x_i, y_i, \theta_i, \sigma_i, \mathbf{d}_i)$  representing x and y coordinates, dominant orientation, scale and SIFT descriptor respectively. Similar to [9], features are densely sampled from single scale and are rotationally variant, thus  $\theta$  and  $\sigma$  are ignored. From a randomly sampled set of the descriptors, a codebook of size  $K$ ,  $\{c_k\}_K$  is learned using  $K$ -means clustering. Then, every feature descriptor  $\mathbf{d}_i$  is mapped to the closest cluster  $c_k$  in the feature space measured by Euclidean distance. To form a local pair of visual words, two visual words must be within  $\gamma$  pixels away from each other spatially and such pair is represented as:

$$f_{(i,j)} = \left( \frac{(x_i + x_j)}{2}, \frac{(y_i + y_j)}{2}, (c_i, c_j) \right) \tag{1}$$

The ordering of  $i$  and  $j$  is determined by  $c_i \leq c_j$  which is just a comparison between cluster indices. Given a set of visual word pairs formed for an image  $I$ , we establish a histogram  $H_I$  where each bin stores the occurrence of a particular pair of visual words  $(c_i, c_j)$ . Once histograms are generated for all training images, a kernel is computed. This is further discussed in Sect. 3.4.

Given  $K$  number of clusters, there are  $K \times (K + 1)/2$  number of pairwise feature clusters which can potentially be large even if the size of  $K$  is moderate. In the case of [8], they set  $K$  to be 1000, so the number of possible pairs is around 500K. While feature selection [8,11] can be used to reduce the number of such clusters, it is often done in a supervised manner. This precisely motivates our work on Local Pairwise Codebook (LPC) as we learn a compact set of pairwise feature clusters in an unsupervised manner and its codebook size is not governed by the number of single feature clusters. Although supervised feature selection may of course be applied to LPC to further increase discrimination if possible, we will not discuss this in the paper. We introduce LPC in the next section.

### 3.2 Pairing Spatially Close Feature Descriptors

As described previously, each key-point  $f_i$  in an image is represented as a tuple of  $(x_i, y_i, \mathbf{d}_i)$ . In LPC, we first pair up features which are within  $\gamma$  pixels away from each other. Each of such pairs, also referred to as local pairwise feature, is encoded as follows:

$$f_{(i,j)} = \left( \frac{(x_i + x_j)}{2}, \frac{(y_i + y_j)}{2}, [\mathbf{d}_i \mathbf{d}_j] \right) \quad (2)$$

The ordering of  $i$  and  $j$  can be achieved by any total order to ensure order invariance when descriptors are paired. In the case of our work, we determine such ordering by comparing the values of the first non-equal dimension of two descriptors  $\mathbf{d}_i$  and  $\mathbf{d}_j$  encountered. For this ordering to be robust against noise, we have applied discretization on each descriptor by  $\lceil \mathbf{d}_i / \delta \rceil$  where  $\delta$  is a small constant. Once the descriptors are concatenated as  $\mathbf{d}_{ij}$ , it can be viewed as a data point in the joint feature space. We use  $[\mathbf{d}_i \mathbf{d}_j]$  and  $\mathbf{d}_{ij}$  interchangeably. After the descriptor pairing process,  $K$ -means is applied on a random collection of local pairwise feature descriptors to learn a codebook  $C = \{c_k\}_K$  where  $K$  denotes the codebook size. Then, for every image, a histogram where each bin represents occurrences of one local pairwise codeword  $c_k$  is established. An illustrated example of how our approach works is given in Fig. 2.

The construction of LPC is done independently from the single feature codebook. Thus, the size of LPC is directly controlled by  $K$ -means and feature selection is not required if  $K$  is set to a moderate size. Since the distribution of the local pairwise features is considered explicitly during clustering, the clusters are likely to capture local structure information, specifically correlation between spatially close descriptors. Therefore, compared to quadratic growth in the number of pairwise feature clusters, LPC is a *relatively compact* and *general* codebook that can be learned unsupervised.

### 3.3 Efficient Pairwise Cluster Assignment

With LPC, cluster assignment is done on a paired descriptor  $\mathbf{d}_{ij}$  instead of  $\mathbf{d}_i$  and  $\mathbf{d}_j$  individually. Given  $N$  descriptors of  $D$  dimension in an image, if  $P$  number of pairs are formed per descriptor on average, then the time complexity of finding the nearest clusters for  $N \times P$  paired descriptors by naïvely computing their Euclidean distances to  $K$  clusters would be  $O(PDNK)$ . When compared to using the single feature codebook, it grows linear in  $P$ . While hierarchical  $K$ -means [13] or  $K$ -means with approximate nearest neighbor search [14] might be an attractive solution to efficiently look for the nearest cluster, they are designed to reduce  $O(K)$  to  $O(\log(K))$  with different approximations and are more suitable in image retrieval tasks where a large codebook is required to increase discrimination. Here, we outline an exact and efficient nearest neighbor search algorithm specific to LPC. It is based on the observation that data redundancy is present in the pairs of descriptors formed within an image. The resulting algorithm decouples  $P$  and  $D$  and its time complexity becomes  $O((P + D)NK)$ . Thus, with high dimensional descriptors like SIFT and a moderate number of pairs formed (e.g. 50), the efficiency of LPC is greatly improved.

In order to search for the nearest cluster of a paired descriptor  $\mathbf{d}_{ij}$ , we need to compute its squared Euclidean distance to each cluster  $\mathbf{c}_k$  and find the one with the minimum distance. Since each pairwise feature cluster is an exemplar of local pairwise features, its vector  $\mathbf{c}_k$  can be seen as a concatenation of two single feature descriptor exemplars  $[\mathbf{c}_{k1} \mathbf{c}_{k2}]$ . Thus, the squared distance between  $\mathbf{c}_k$  and  $\mathbf{d}_{ij}$  can be rewritten as a sum of two squared distances given in (4). Since both  $\mathbf{d}_i$  and  $\mathbf{d}_j$  are likely to be paired again with some other descriptors within their local neighborhood, we can cache  $\|\mathbf{c}_{k1} - \mathbf{d}_i\|^2$  and  $\|\mathbf{c}_{k2} - \mathbf{d}_j\|^2$  into matrices  $Q^1$  and  $Q^2$  respectively so to avoid recomputing these partial distances as stated in (5). Both matrices have the size of  $K \times N$  and are indexed by cluster and descriptor indices. Thus, if we have the two matrices calculated prior to the pairwise feature cluster assignment, then the distance calculation for each pair becomes just a sum of two values in the matrices.

$$\|\mathbf{c}_k - \mathbf{d}_{ij}\|^2 = \|[\mathbf{c}_{k1} \mathbf{c}_{k2}] - [\mathbf{d}_i \mathbf{d}_j]\|^2 \quad (3)$$

$$= \|\mathbf{c}_{k1} - \mathbf{d}_i\|^2 + \|\mathbf{c}_{k2} - \mathbf{d}_j\|^2 \quad (4)$$

$$= Q_{k1,i}^1 + Q_{k2,j}^2 \quad (5)$$

Computation of the matrices  $Q^1$  and  $Q^2$  takes  $O(DNK)$  time as distances between  $N$  descriptors and  $2K$  single feature descriptor exemplars are calculated. Then, we have  $N \times P$  descriptor pairs to look up the distances to  $K$  clusters to find the closest one and this takes  $O(PNK)$  time. By combining the two operations, we get the time complexity of  $O((P + D)NK)$ . In the case of our experiments, we have used  $P = 48$  ( $\gamma = 24$ ) and  $D = 32$  by default and the efficiency of LPC has increased by 20 folds using this proposed technique.

### 3.4 Computing Kernel

Once a codebook is learned, a histogram or its variant is established for each image to compute either histogram intersection kernel (HIK) or spatial pyramid matching kernel (SPMK) [9] for classification. For a pair of  $\ell_1$  normalized histograms  $H_X$  and  $H_Y$  representing images  $X$  and  $Y$  respectively, HIK is computed as given in (6) where  $i$  denotes a histogram bin index.

$$K_{HIK}(H_X, H_Y) = \sum_{i=1}^K \min(H_X(i), H_Y(i)) \quad (6)$$

For SPMK, each image is partitioned into  $M \times M$  equal-sized regions over multiple levels of resolution. At each level  $l$ , by defining the resolution  $M$  to be  $2^l$ ,  $M \times M$  histograms are established and concatenated to produce  $H^l$ . Constructing such histograms essentially captures global spatial information at different granularity. Finally, (7) computes the similarity between two histograms  $H_X$  and  $H_Y$  where we set the maximum level  $L$  to be 2.

$$K_{SPM}^L(H_X, H_Y) = \frac{1}{2^L} K_{HIK}(H_X^0, H_Y^0) + \sum_{l=1}^L \frac{1}{2^{L-l+1}} K_{HIK}(H_X^l, H_Y^l) \quad (7)$$

## 4 Experimental Results

This section presents the experimental results of LPC on five challenging datasets. For all datasets, SIFT descriptors (32 dimension,  $2 \times 2$  grids, 8 orientation bins) are densely sampled at every 8 pixel step and are described from  $16 \times 16$  patches. In [20], their experimental results have shown that SIFT descriptors with  $2 \times 2$  grids perform competitively against the one with  $4 \times 4$  grids. We have also conducted our own experiments on the datasets and verified that the  $2 \times 2$  performs competitively against the  $4 \times 4$ . By default, the neighborhood threshold  $\gamma$  is set to 24 pixels which forms 48 pairs per descriptor on average. We use the following baselines to compare against our LPC:

1. **QPC**: Quadratic Pairwise Codebook as described in Sect. 3.1. We have evaluated with  $K = \{80, 200\}$  resulting in  $\{3240, 20100\}$  number of pairwise feature clusters.
2. **QPC+Sel**: Quadratic Pairwise Codebook with frequency-based feature selection outlined in [8]. 1000 most frequently appearing pairwise feature clusters are chosen from each category.
3. **Sgl**: A single feature codebook approach. This baseline is our reimplementation of Lazebnik et. al. [9].

In fact, we have also tried other methods to compare against LPC, such as applying supervised feature aggregation [4] to compress QPC. However, due to a large number of categories to be learned and a relatively few training examples available, it has resulted in a performance similar to QPC+Sel. Thus, we report

the results of the above three baselines only. To construct the single feature codebook and LPC for each dataset, 100K and 250K descriptor samples are used respectively. For all experiments, we have repeated eight times with different sets of training and testing images randomly sampled.

#### 4.1 Scene Categorization

For scene categorization, we have used the 15 Scenes [9] and 67 Indoors [16] datasets. 15 Scenes contains images of 15 categories ranging from indoor scenes (e.g. living room and kitchen) to outdoor scenes (e.g. mountain and street). 100 images per category are used for training and the rest is used for testing. It is clear from the results shown in Table 1 that LPC outperforms Sgl and QPC(+Sel) in both HIK and SPMK. Although QPC has performed better than Sgl in the case of HIK, it has failed to do so in SPMK. QPC+Sel has selected 2562 and 4400 feature clusters from  $K = 80$  (3240) and  $K = 200$  (20100) respectively, but with no improvement observed. While the performance of Sgl reaches its peak when  $K$  is 1600 and 400 for HIK and SPMK respectively, LPC continues to improve as we increase  $K$  in both settings. For 67 Indoors dataset, the same experimental setup in [16] is used. In general, as similar to the results of 15 Scenes, LPC performs better in both HIK and SPMK settings.

**Table 1.** Recognition accuracy (%) on 15 Scenes and 67 Indoors. HIK refers to histogram intersection kernel and SPMK refers to spatial matching kernel.

Methods	K	15 Scenes		67 Indoors	
		HIK	SPMK	HIK	SPMK
QPC	80	76.61±0.44	<b>81.41±0.56</b>	27.28±1.44	34.83±0.97
	200	<b>78.25±0.45</b>	80.93±0.36	30.75±1.38	<b>36.15±0.95</b>
QPC+Sel	80	<b>76.03±0.46</b>	<b>81.11±0.55</b>	26.96±1.76	<b>34.27±0.93</b>
	200	74.69±0.60	78.71±0.60	<b>27.29±0.60</b>	32.91±0.60
Sgl	400	74.75±0.66	<b>81.76±0.47</b>	23.89±0.92	33.82±1.10
	800	75.62±0.58	81.50±0.48	26.11±0.98	34.82±0.83
	1600	<b>76.67±0.61</b>	81.40±0.42	27.67±1.03	<b>35.13±1.07</b>
	3200	76.13±0.51	80.34±0.37	<b>28.77±1.85</b>	34.71±1.13
LPC	800	76.78±0.53	82.50±0.63	27.30±1.44	35.80±0.86
	1600	78.42±0.23	83.07±0.50	29.71±0.70	37.16±1.20
	3200	<b>79.76±0.38</b>	<b>83.40±0.58</b>	<b>32.35±0.68</b>	<b>38.36±0.63</b>

#### 4.2 Caltech-101/256 Datasets

The Caltech-101 dataset [3] comprises of 101 different object classes. In our experiments, we have used 30 images per category for training and the rest for testing, excluding Background class. For an efficiency reason, we have down-sized images preserving their aspect ratios if their longer sides are greater than 1000 pixels. The results are given in Table 2. Compared with the results of



**Table 2.** Recognition accuracy (%) on Caltech-101 and Caltech-256

Methods	K	Caltech-101		Caltech-256	
		HIK	SPMK	HIK	SPMK
QPC	80	51.91±1.20	<b>67.27±1.00</b>	23.12±0.42	31.50±0.52
	200	<b>54.68±0.63</b>	66.04±0.76	<b>25.57±0.43</b>	<b>32.13±0.37</b>
QPC+Sel	80	<b>51.87±1.20</b>	<b>67.16±0.93</b>	23.17±0.30	<b>31.10±0.35</b>
	200	51.04±0.89	63.34±0.96	<b>26.18±0.62</b>	30.48±0.66
Sgl	400	45.40±0.80	<b>65.38±1.16</b>	18.67±0.42	29.06±0.26
	800	47.38±0.67	64.45±0.44	19.10±0.49	<b>29.14±0.38</b>
	1600	47.74±0.40	62.88±0.83	19.63±0.47	28.44±0.35
	3200	<b>47.95±0.97</b>	60.25±1.09	<b>20.00±0.46</b>	27.75±0.41
LPC	800	53.06±0.99	69.90±0.48	24.69±0.34	33.93±0.54
	1600	55.50±0.82	70.48±0.73	26.33±0.37	35.08±0.47
	3200	<b>57.08±0.88</b>	<b>71.00±0.48</b>	<b>28.11±0.38</b>	<b>35.74±0.41</b>

15 Scenes, the performance boost for LPC over Sgl is much more obvious - improving the results by around 9% and 6% for HIK and SPMK respectively. We suspect that objects tend to have more local structures present than scenes and LPC has exploited these to obtain boost in the recognition. Overall, LPC has outperformed both Sgl and QPC(+Sel) showing similar trends seen in the previous experiment.

The Caltech-256 dataset [6] succeeds the Caltech-101 dataset with several improvements including increased intra-class variability and variability in object pose and location. We have used 30 training and 25 testing images per category, excluding Clutter class. We have down-sized images if their longer sides are greater than 300 pixels. The results are presented in Table 2. The performance increase of LPC over Sgl is akin to the previous results. Interestingly, the best result obtained with LPC HIK ( $K = 3200$ ) is close to Sgl SPMK ( $K = 800$ ). Since the number of bins required for LPC HIK is 3200 and Sgl SPMK is 16800 which is 5 times more, this implies the effectiveness of exploiting local structures over global structure when variability of object pose and location is high. Of course, LPC SPMK, the combination of both local and global information, has shown to perform the best on this dataset as well.

### 4.3 MSRCv2 Dataset

The MSRCv2 dataset is a relatively small object class database compared to Caltech-101/256, but is considered be much more difficult dataset due to its high intra-class variability [22]. We have simply followed the experimental setup used in [25], except we have used dense sampling instead of an interest point detector to extract feature descriptors. Nine out of fifteen classes are chosen (i.e. cow, airplanes, faces, cars, bikes, books, signs, sheep and chairs) where each class contains 30 images. For each experiment, we have randomly sampled 15 training and 15 testing images class and no background is removed from the images. We have used HIK in this experiment. LPC has performed better

against the baselines with an accuracy of  $83.9 \pm 2.9\%$  with  $K$  being 3200. The highest accuracy obtained by each baseline is as follows: QPC ( $81.8 \pm 3.4\%$ ), QPC+Sel ( $80.8 \pm 3.4\%$ ) and Sgl ( $81.7 \pm 2.8\%$ ). Our results are higher than the results reported in [25] where  $2^{nd}$  (pairwise) and  $10^{th}$  order spatial features have obtained accuracies of  $78.3 \pm 2.6\%$  and  $80.4 \pm 2.5\%$  respectively.

#### 4.4 Comparison with Large Patch SIFT Descriptors

We have experimented the baseline Sgl with three different configurations of SIFT descriptor to be sampled at every 8 pixels, i.e.  $24 \times 24$  patch with  $3 \times 3$  grids ( $D = 72$ ),  $32 \times 32$  patch with  $4 \times 4$  grids ( $D = 128$ ),  $40 \times 40$  patch with  $5 \times 5$  grids ( $D = 200$ ). These try to mimic the effect of LPC with different neighborhood thresholds, i.e.  $\gamma = \{8, 16, 24\}$ . For an example, with  $40 \times 40$  patch SIFT descriptors, they potentially cover all possible pairings of  $16 \times 16$  SIFT descriptors with  $\gamma = 24$ . The results are presented in Table 3. For all datasets, these large patch SIFT descriptors improve over the Sgl baseline evaluated earlier. However, LPC performs better across all datasets using just  $16 \times 16$  patch SIFT descriptors, especially, the performance gap is significant for Caltech-256 which is considered to be a hard dataset. Not only this implies the robustness of LPC, it is also relatively more efficient compared to these baselines if time required to do feature extraction, codebook construction and cluster assignment is considered. In fact, it is possible to apply LPC on larger patches as well. For an instance, LPC with  $24 \times 24$  patch with  $2 \times 2$  grids ( $D = 32$ ) has achieved  $72.15 \pm 0.68\%$  accuracy on Caltech-101.

**Table 3.** Recognition accuracy (%) of Sgl with large patch SIFT descriptors. Although the performance of Sgl has increased, LPC that uses  $16 \times 16$  patches still consistently performs better than these baselines.

Methods	K	15 Scenes	67 Indoors	Caltech-101	Caltech-256
Sgl (24x,72D)	800	81.96±0.44	36.56±0.78	<b>67.34±1.11</b>	31.32±0.34
	1600	<b>82.25±0.65</b>	37.42±1.23	66.87±0.96	<b>31.50±0.41</b>
	3200	81.77±0.43	<b>37.57±0.85</b>	66.07±0.42	31.14±0.49
Sgl (32x,128D)	800	81.25±0.59	36.04±0.75	<b>69.18±0.86</b>	31.57±0.48
	1600	<b>81.74±0.80</b>	36.08±0.87	68.82±0.73	<b>32.26±0.21</b>
	3200	81.17±0.74	<b>36.86±0.92</b>	68.38±0.66	32.22±0.43
Sgl (40x,200D)	800	80.40±0.68	34.23±1.66	68.95±1.26	31.08±0.46
	1600	<b>80.75±0.48</b>	<b>35.38±1.08</b>	<b>69.47±0.69</b>	31.52±0.43
	3200	80.61±0.48	35.08±0.85	69.40±1.00	<b>31.87±0.25</b>
LPC	3200	<b>83.40±0.58</b>	<b>38.36±0.63</b>	<b>71.00±0.48</b>	<b>35.74±0.41</b>

#### 4.5 Comparison with Different Neighborhood Thresholds $\gamma$

In this section, we have also experimented LPC with different neighborhood thresholds  $\gamma = \{8, 16, 24, 32\}$  using SPMK. Table 4 shows the results on the first four datasets. Overall, with any  $\gamma$  and codebook size tried, it has improved over

**Table 4.** The performance comparison with different  $\gamma$ . Each  $\gamma$  shows the average number of pairs formed per descriptor in brackets.

Dataset	K	$\gamma$ (#pairs)			
		8 (8)	16 (24)	24 (48)	32 (80)
15 Scenes	1600	82.83±0.62	82.92±0.44	83.07±0.50	82.84±0.52
	3200	82.68±0.47	83.18±0.63	<b>83.40±0.58</b>	83.28±0.49
67 Indoors	1600	37.29±0.96	37.76±1.43	37.16±1.20	36.81±1.38
	3200	37.46±1.43	<b>38.64±1.32</b>	38.36±0.63	37.75±0.67
Caltech-101	1600	68.49±0.91	69.92±0.64	70.48±0.80	70.89±0.84
	3200	67.95±0.76	70.08±0.56	71.00±0.48	<b>71.05±0.85</b>
Caltech-256	1600	32.69±0.40	34.28±0.39	35.08±0.47	34.68±0.43
	3200	32.75±0.52	34.80±0.71	<b>35.74±0.41</b>	35.57±0.52

**Table 5.** The performance comparison with different step sizes

Dataset	K	step size		
		4	6	8
15 Scenes	1600	81.31±0.38	81.39±0.47	83.07±0.50
	3200	81.97±0.45	82.00±0.43	<b>83.40±0.58</b>
67 Indoors	1600	37.58±1.04	37.98±1.21	37.16±1.20
	3200	38.93±0.48	<b>39.63±0.69</b>	38.36±0.63
Caltech-101	1600	72.40±0.79	71.65±0.52	70.48±0.73
	3200	<b>72.94±0.54</b>	72.01±0.49	71.00±0.48
Caltech-256	1600	36.85±0.56	36.06±0.54	35.08±0.47
	3200	<b>37.71±0.47</b>	37.10±0.46	35.74±0.41

Sgl (c.f. Tables 1 and 2). This shows the benefit of capturing local interaction between feature descriptors. For 15 Scenes, any  $\gamma$  larger than 8 pixels achieve similar performance. On the contrary, with Caltech-101 and Caltech-256, the performance tends to improve as  $\gamma$  is increased for any codebook size tried.

#### 4.6 Comparison with Different Step Sizes

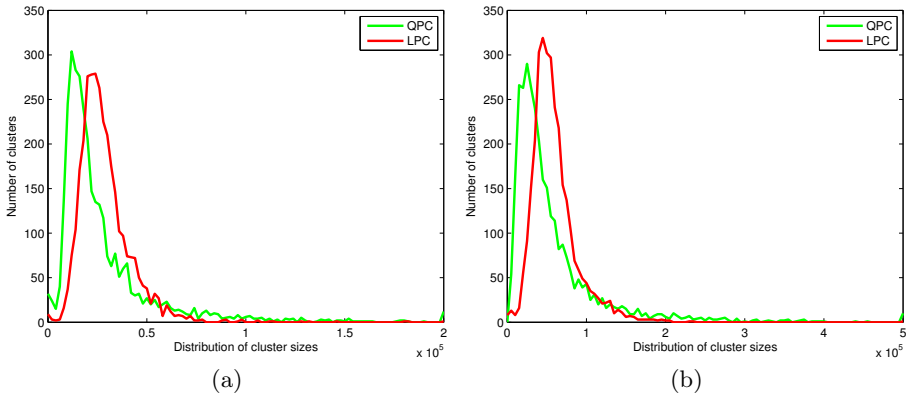
In this section, LPC is evaluated on different step sizes used by dense sampling. The neighborhood threshold  $\gamma$  is set to 24 pixels. We report the results on SPMK in Table 5. For both Caltech-101 and Caltech-256, the performance increases as the step size gets smaller.

#### 4.7 Comparison with Previously Published Results

This section compares LPC with the previously published results which were or are known to be the state-of-the-art for the datasets. For each dataset, we report the highest recognition accuracy of LPC obtained from the previous sections. For the comparison to be fair, we have only included results that are obtained from using a single descriptor or cue. As shown in Table 6, LPC has performed

**Table 6.** Performance comparison with the state-of-the-art methods based on a single descriptor. Results for a single image scale is reported wherever possible.

Methods	15 Scenes	67 Indoors	Caltech-101	Caltech-256
KC [5]	76.67±0.39	-	64.14±1.18	27.17±0.46
SPM [9]	81.40±0.50	-	64.60±0.80	-
KSPM [6]	-	-	67.40	34.10
NBNN [1]	-	-	70.40	-
HC [23]	82.30±0.49	-	-	-
HG [26]	<b>85.20</b>	-	73.10	-
ScSPM [24]	80.28±0.93	-	<b>73.20±0.54</b>	34.02±0.35
ROI+Gist [16]	-	>30.00	-	-
LPC	83.40±0.58	<b>39.63±0.69</b>	72.94±0.54	<b>37.71±0.47</b>

**Fig. 3.** Distributions of cluster sizes estimated on (a) 15 Scenes and (b) Caltech-101

competitively against other methods across all datasets. For Caltech-256, it has outperformed the state-of-the-art methods.

#### 4.8 On the Distribution of Local Pairwise Features

In this section, we have estimated the distributions of the cluster sizes for the traditional approaches (QPC) and our approach (LPC) using the 15 Scenes and Caltech-101 datasets. To be unbiased, we have used roughly the same number of clusters for both approaches, i.e.  $K = 80$  (3240 pairwise clusters) for QPC and  $K = 3200$  for LPC. We have extracted pairs of features from all images and used them to plot the distributions. In total, approximately 90 million and 190 million such features are extracted for 15 Scenes and Caltech-101 respectively.

As depicted in Fig. 3, the distributions of QPC plotted in green has an earlier peak than LPC plotted in red. This means that QPC has placed a lot of its clusters in regions where there are not many data points. Also, the distributions are heavy-tailed due to many points assigned to only a few number of clusters. In contrast, the estimated distributions obtained by LPC seem to be tighter than

the former distributions. Based on this observation, we can infer that LPC has explicitly considered the underlying distribution of the local pairwise features and balanced its cluster sizes as much as possible. Therefore, we believe that this observation supports our intuition presented earlier with Fig. 11.

## 5 Conclusion

In this paper, we have presented a simple, yet effective method of building a compact codebook that encodes local spatial information with joint feature space clustering called Local Pairwise Codebook. For it being a simple method, LPC has outperformed the methods with which we have compared, and performed competitively against the previously published results. Our future work involves building LPC based on interest point detectors instead of the dense sampling strategy used in our experiments as well as incorporating additional spatial information like direction and distance.

## Acknowledgments

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

## References

1. Boiman, O., Shechtman, E., Irani, M.: In defence of Nearest-Neighbor based image classification. In: CVPR (2008)
2. Chum, O., Perdoch, M., Matas, J.: Geometric min-Hashing: Finding a (thick) needle in a haystack. In: CVPR (2009)
3. Fei-Fei, L., Fergus, R., Perona, P.: Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In: CVPR Workshop (2004)
4. Fulkerson, B., Vedaldi, A., Soatto, S.: Localizing Objects with Smart Dictionaries. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 179–192. Springer, Heidelberg (2008)
5. van Gemert, J., Veenman, C., Geusebroek, J.: Visual Word Ambiguity. In: TPAMI (2010)
6. Griffin, G., Holub, A., Perona, P.: Caltech 256 object category dataset. Technical Report UCB/CSD-04-1366, California Institute of Technology (2006)
7. Lan, X., Zitnick, C., Szeliski, R.: Local Bi-gram Model for Object Recognition MSR-TR-2007-54. Technical Report, Microsoft Research (2007)
8. Lazebnik, S., Schmid, C., Ponce, J.: A Maximum Entropy Framework for Part-Based Texture and Object Recognition. In: ICCV (2005)
9. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR (2006)
10. Ling, H., Soatto, S.: Proximity Distribution Kernels for Geometric Context in Category Recognition. In: ICCV (2007)

11. Liu, D., Hua, G., Viola, P., Chen, T.: Integrated feature selection and higher-order spatial feature extraction for object categorization. In: CVPR (2008)
12. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. In: IJCV (2004)
13. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: CVPR (2006)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
15. Quak, T., Ferrari, V., Leibe, B., Gool, L.V.: Efficient Mining of Frequent and Distinctive Feature Configurations. In: ICCV (2007)
16. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR (2009)
17. Savarese, S., Winn, J., Criminisi, A.: Discriminative Object Class Models of Appearance and Shape by Correlations. In: CVPR (2006)
18. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering objects and their location in images. In: ICCV (2005)
19. Torresani, L., Szummer, M., Fitzgibbon, A.: Learning query-dependent prefilters for scalable image retrieval. In: CVPR (2009)
20. Uijlings, J., Smeulders, A., Scha, R.: Real-time Bag of Words. In: CIVR (2009)
21. Vedaldi, A., Soatto, S.: Relaxed matching kernels for robust image comparison. In: CVPR (2008)
22. Winn, J., Criminisi, A., Minka, T.: Object Categorization by Learned Universal Visual Dictionary. In: CVPR (2005)
23. Wu, J., Rehg, J.: Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2009)
24. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
25. Yimeng, Z., Tsuhan, C.: Efficient Kernels for identifying unbounded-order spatial features. In: CVPR (2009)
26. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical Gaussianization for Image Classification. In: ICCV (2009)

# Image-to-Class Distance Metric Learning for Image Classification

Zhengxiang Wang, Yiqun Hu, and Liang-Tien Chia

Center for Multimedia and Network Technology, School of Computer Engineering  
Nanyang Technological University, 639798, Singapore  
wang0460@ntu.edu.sg, yiqun.hu@gmail.com, asltchia@ntu.edu.sg

**Abstract.** Image-To-Class (I2C) distance is first used in Naive-Bayes Nearest-Neighbor (NBNN) classifier for image classification and has successfully handled datasets with large intra-class variances. However, the performance of this distance relies heavily on the large number of local features in the training set and test image, which need heavy computation cost for nearest-neighbor (NN) search in the testing phase. If using small number of local features for accelerating the NN search, the performance will be poor.

In this paper, we propose a large margin framework to improve the discrimination of I2C distance especially for small number of local features by learning Per-Class Mahalanobis metrics. Our I2C distance is adaptive to different class by combining with the learned metric for each class. These multiple Per-Class metrics are learned simultaneously by forming a convex optimization problem with the constraints that the I2C distance from each training image to its belonging class should be less than the distance to other classes by a large margin. A gradient descent method is applied to efficiently solve this optimization problem. For efficiency and performance improved, we also adopt the idea of spatial pyramid restriction and learning I2C distance function to improve this I2C distance. We show in experiments that the proposed method can significantly outperform the original NBNN in several prevalent image datasets, and our best results can achieve state-of-the-art performance on most datasets.

## 1 Introduction

Image classification is a highly useful yet still challenging task in computer vision community due to the large intra-class variances and ambiguities of images. Many efforts have been done for dealing with this problem and they can roughly be divided into learning-based and non-parametric methods according to [1]. Compared to learning-based methods, non-parametric methods directly classify on the test set and do not require any training phase. So in most cases, learning-based methods can achieve better recognition performance than non-parametric methods as they have learned the model from the training set, which is useful for classifying test images. But recently a new non-parametric method named as NBNN *et al.* [2] was proposed, which reported comparable performance to those

top learning-based methods. They contribute such achievement to the avoidance of descriptor quantization and the use of Image-To-Class (I2C) distance instead of Image-To-Image (I2I) distance, since they proved descriptor quantization and I2I distance lead to significant degradation for classification.

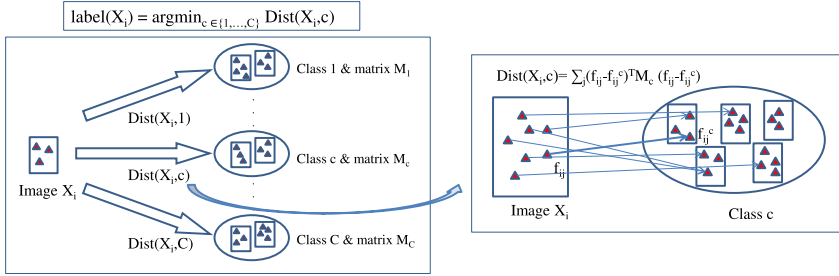
However, the performance of this I2C distance relies heavily on the large number of local features in the training set and test image. For example, the state-of-the-art performance they reported in Caltech 101 dataset is achieved by densely sampling large redundant local features for both training and test images, which results in about 15000 to 20000 features per image. Such large number of features makes the nearest-neighbor (NN) search in I2C distance calculation computationally expensive when classifying a test image, which limits its scalability in real-world application. If only small number of local features is used, the performance of this I2C distance will be poor as shown in the later experiment section, although it needs less testing time.

In this paper, we aim to enhance the performance of I2C distance especially for small number of local features, so as to speed up the testing phase while maintaining excellent result. To achieve this, we propose a training phase to exploit the training information and suggest a distance metric learning method for the I2C distance. Our method avoids the shortcoming of both non-parametric methods and most learning-based methods involving I2I distance and descriptor quantization. This leads to a better recognition performance than NBNN and those learning-based methods. For each class, we learn the class specific Mahalanobis metric to combine with the corresponding I2C distance. When classifying a test image, we select the shortest I2C distance among its Mahalanobis I2C distances to all classes as its predicted class label. Since only the metric of the belonging class can well characterize the local features of the test image, such Per-Class metrics can better preserve the discriminate information between different classes compared to a single global metric.

We adopt the idea of large margin from SVM in our optimization problem for learning these Per-Class metrics, which is also used for distance metric learning by Weinberger *et al.* [18] recently. For each training image, we separate the I2C distance to its belonging class from those to any other class by a large margin, and form a large margin convex optimization problem as an instance of semi-definite programming (SDP). Then we apply an efficient gradient descent method to solve this optimization problem. We also show the incremental learning ability of our Per-Class metric learning, which enables our method to be used for on-line learning and it can be easily scaled-up for handling large number of classes. Figure 1 gives the illustration of our classification structure. Notations used in the figure will be explained in Section 2.1. Compared to NBNN classifier, the main difference is that we use Mahalanobis distance with our learned Per-Class metrics instead of Euclidean distance in NBNN, while the way to classify a test image is similar.

Moreover, we adopt the idea of spatial pyramid match [9] and learning I2C distance function [16] to generate a more discriminative distance for improving classification accuracy. Since the main computation burden is the NN search in





**Fig. 1.** The classification structure of our method. The rectangular and triangles denote an image and its local feature points respectively. The ellipse denotes a class with images (rectangular) inside it. The I2C distance from image  $X_i$  to a class  $c$  is formed by the sum of Mahalanobis distance between each local feature  $f_{ij}$  and its NN  $f_{ij}^c$  in class  $c$  with the matrix  $M_c$  learned for that class. The predicted label of image  $X_i$  is chosen by selecting the shortest I2C distance. Section 2.1 gives a full explanation of these notations.

I2C distance calculation rather than metric learning, we also propose an acceleration method using spatial restriction for speeding up the NN search, which can preserve or even improve the classification accuracy in most datasets. Our objectives for improving the I2C distance are twofold: minimizing the testing time and improving the classification performance.

We describe our large margin optimization problem as well as an efficient solver in Section 2, where we also discuss our improvements in addition to the learned metrics. We evaluate our method and compare it with other methods in Section 3. Finally, we conclude this paper in Section 4.

## 2 Distance Metric Learning for I2C Distance

In this section, we formulate a large margin convex optimization problem for learning the Per-Class metrics and introduce an efficient gradient descent method to solve this problem. We also adopt two strategies to further enhance the discrimination of our learned I2C distance.

### 2.1 Notation

Our work deals with the image represented by a collection of its local feature descriptors extracted from patches around each keypoint. So let  $F_i = \{f_{i1}, f_{i2}, \dots, f_{im_i}\}$  denote features belonging to image  $X_i$ , where  $m_i$  represents the number of features in  $X_i$  and each feature is denoted as  $f_{ij} \in R^d, \forall j \in \{1, \dots, m_i\}$ . To calculate the I2C distance from an image  $X_i$  to a candidate class  $c$ , we need to find the NN of each feature  $f_{ij}$  from class  $c$ , which is denoted as  $f_{ij}^c$ . The original I2C distance from image  $X_i$  to class  $c$  is defined as the sum

of Euclidean distances between each feature in image  $X_i$  and its NN in class  $c$  and can be formulated as:

$$Dist(X_i, c) = \sum_{j=1}^{m_i} \| f_{ij} - f_{ij}^c \|^2 \tag{1}$$

After learning the Per-Class metric  $M_c \in R^{d \times d}$  for each class  $c$ , we replace the Euclidean distance between each feature in image  $X_i$  and its NN in class  $c$  by the Mahalanobis distance and the learned I2C distance becomes:

$$Dist(X_i, c) = \sum_{j=1}^{m_i} (f_{ij} - f_{ij}^c)^T M_c (f_{ij} - f_{ij}^c) \tag{2}$$

This learned I2C distance can also be represented in a matrix form by introducing a new term  $\Delta X_{ic}$ , which is a  $m_i \times d$  matrix representing the difference between all features in the image  $X_i$  and their nearest neighbors in the class  $c$  formed as:

$$\Delta X_{ic} = \begin{pmatrix} (f_{i1} - f_{i1}^c)^T \\ (f_{i2} - f_{i2}^c)^T \\ \dots \\ (f_{im_i} - f_{im_i}^c)^T \end{pmatrix} \tag{3}$$

So the learned I2C distance from image  $X_i$  to class  $c$  can be reformulated as:

$$Dist(X_i, c) = Tr(\Delta X_{ic} M_c \Delta X_{ic}^T) \tag{4}$$

This is equivalent to the equation (2). If  $M_c$  is an identity matrix, then it's also equivalent to the original Euclidean distance form of equation (1). In the following subsection, we will use this formulation in the optimization function.

### 2.2 Problem Formulation

The objective function in our optimization problem is composed of two terms: the regularization term and error term. This is analogous to the optimization problem in SVM. In the error term, we incorporate the idea of large margin and formulate the constraint that the I2C distance from image  $X_i$  to its belonging class  $p$  (named as positive distance) should be smaller than the distance to any other class  $n$  (named as negative distance) with a margin. The formula is given as follows:

$$Tr(\Delta X_{in} M_n \Delta X_{in}^T) - Tr(\Delta X_{ip} M_p \Delta X_{ip}^T) \geq 1 \tag{5}$$

In the regularization term, we simply minimize all the positive distances similar to [20]. So for the whole objective function, on one side we try to minimize all the positive distances, on the other side for every image we keep those negative distances away from the positive distance by a large margin. In order to allow

soft-margin, we introduce a slack variable  $\xi$  in the error term, and the whole convex optimization problem is therefore formed as:

$$\begin{aligned}
 \min_{M_1, M_2, \dots, M_C} O(M_1, M_2, \dots, M_C) &= & (6) \\
 (1 - \lambda) \sum_{i, p \rightarrow i} Tr(\Delta X_{ip} M_p \Delta X_{ip}^T) + \lambda \sum_{i, p \rightarrow i, n \rightarrow i} \xi_{ipn} \\
 s.t. \forall i, p, n : Tr(\Delta X_{in} M_n \Delta X_{in}^T) - Tr(\Delta X_{ip} M_p \Delta X_{ip}^T) &\geq 1 - \xi_{ipn} \\
 \forall i, p, n : \xi_{ipn} &\geq 0 \\
 \forall c : M_c &\succeq 0
 \end{aligned}$$

This optimization problem is an instance of SDP, which can be solved using standard SDP solver. However, as the standard SDP solvers is computation expensive, we use an efficient gradient descent based method derived from [20,19] to solve our problem. Details are explained in the next subsection.

### 2.3 An Efficient Gradient Descent Solver

Due to the expensive computation cost of standard SDP solvers, we propose an efficient gradient descent solver derived from Weinberger *et al.* [20,19] to solve this optimization problem. Since the method proposed by Weinberger *et al.* targets on solving only one global metric, we modify it to learn our Per-Class metrics. This solver updates all matrices iteratively by taking a small step along the gradient direction to reduce the objective function (6) and projecting onto feasible set to ensure that each matrix is positive semi-definite in each iteration. To evaluate the gradient of objective function for each matrix, we denote the matrix  $M_c$  for each class  $c$  at  $t^{th}$  iteration as  $M_c^t$ , and the corresponding gradient as  $G(M_c^t)$ . We define a set of triplet error indices  $N^t$  such that  $(i, p, n) \in N^t$  if  $\xi_{ipn} > 0$  at the  $t^{th}$  iteration. Then the gradient  $G(M_c^t)$  can be calculated by taking the derivative of objective function (6) to  $M_c^t$ :

$$G(M_c^t) = (1 - \lambda) \sum_{i, c=p} \Delta X_{ic}^T \Delta X_{ic} + \lambda \sum_{(i, p, n) \in N^t, c=p} \Delta X_{ic}^T \Delta X_{ic} - \lambda \sum_{(i, p, n) \in N^t, c=n} \Delta X_{ic}^T \Delta X_{ic} \quad (7)$$

Directly calculating the gradient in each iteration using this formula would be computational expensive. As the changes in the gradient from one iteration to the next are only determined by the differences between the sets  $N^t$  and  $N^{t+1}$ , we use  $G(M_c^t)$  to calculate the gradient  $G(M_c^{t+1})$  in the next iteration, which would be more efficient:

$$\begin{aligned}
 G(M_c^{t+1}) &= G(M_c^t) + \lambda \left( \sum_{(i, p, n) \in (N^{t+1} - N^t), c=p} \Delta X_{ic}^T \Delta X_{ic} - \sum_{(i, p, n) \in (N^{t+1} - N^t), c=n} \Delta X_{ic}^T \Delta X_{ic} \right) \\
 &\quad - \lambda \left( \sum_{(i, p, n) \in (N^t - N^{t+1}), c=p} \Delta X_{ic}^T \Delta X_{ic} - \sum_{(i, p, n) \in (N^t - N^{t+1}), c=n} \Delta X_{ic}^T \Delta X_{ic} \right)
 \end{aligned} \quad (8)$$

Since  $(\Delta X_{ic}^T \Delta X_{ic})$  is unchanged during the iterations, we can accelerate the updating procedure by pre-calculating this value before the first iteration. The

matrix is updated by taking a small step along the gradient direction for each iteration. To enforce the positive semi-definiteness, the updated matrix needs to be projected onto a feasible set. This projection is done by eigen-decomposition of the matrix and truncating all the negative eigenvalues to zeros. As the optimization problem (6) is convex, this solver is able to converge to the global optimum. We summarize the whole work flow in Algorithm 1.

---

**Algorithm 1.** A Gradient Descent Method for Solving Our Optimization Problem

---

Input: step size  $\alpha$ , parameter  $\lambda$  and pre-calculated data  $(\Delta X_{ic}^T \Delta X_{ic}), i \in \{1, \dots, N\}, c \in \{1, \dots, C\}$   
**for**  $c := 1$  to  $C$  **do**  
     $G(M_c^0) := (1 - \lambda) \sum_{i,p \rightarrow i} \Delta X_{ip}^T \Delta X_{ip}$   
     $M_c^0 := I$   
**end for**{Initialize M and gradient for each class}  
Set  $t := 0$   
**repeat**  
    Compute  $N^t$  by checking each error term  $\xi_{ipn}$   
    **for**  $c = 1$  to  $C$  **do**  
        Update  $G(M_c^{t+1})$  using equation (8)  
         $M_c^{t+1} := M_c^t + \alpha G(M_c^{t+1})$   
        Project  $M_c^{t+1}$  for keeping positive semi-definite  
    **end for**  
    Calculate new objective function  
     $t := t + 1$   
**until** Objective function converged  
Output: each matrix  $M_1, \dots, M_C$

---

## 2.4 Scalability and Incremental Learning

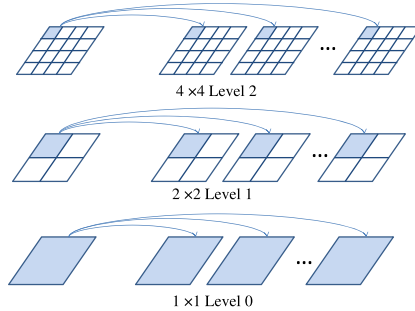
Next we analyze the efficiency of this solver and its scalability. Although the number of triplets is large for dealing with large-scale dataset, for example 151500 triplets in error term for Caltech 101 dataset using 15 images per class for training, we find only a small portion of them are non-zero, which are put into the error index set  $N^t$  and used for updating matrices. To speed up calculating  $N^t$  in each iteration, we also keep an active set of triplets as proposed in [20] for calculating  $N^t$  rather than scanning over all the triplets in each iteration. So this solver runs quickly for updating hundreds of metrics. In our experiment, it needs about 30 iterations to converge for Scene, Sports and Corel datasets, and about 60 iterations for Caltech 101 dataset to converge with an appropriate step size. We can further accelerate the training phase by learning a diagonal matrix for each class, which would alleviate the computation cost especially when there are even more classes, e.g. thousands of classes.

Our method also supports the incremental learning. When new training images of existing class or new class are added, Per-Class metrics do not need to be

re-trained from the beginning. The current learned matrices can be used as initial estimates by changing the identity matrix  $I$  to current matrix for each class in Algorithm 1, and new triplets are added to update all matrices. The updating procedure will converge quickly since pre-learned matrices are relatively close to the optimal. This incremental learning ability shows that our method can be scaled-up to handle large number of classes and support for on-line learning.

## 2.5 More Improvements Based on Mahalanobis I2C Distance

To generate a more discriminative I2C distance for better recognition performance, we improve our learned distance by adopting the idea of spatial pyramid match [9] and learning I2C distance function [16].



**Fig. 2.** The left parallelogram denotes an image, and the right parallelograms denote images in a class. We adopt the idea of spatial pyramid by restricting each feature descriptor in the image to only find its NN in the same subregion from a class at each level.

Spatial pyramid match (SPM) is proposed by Lazebnik *et al.* [9] which makes use of spatial correspondence, and the idea of pyramid match is adapted from Grauman *et al.* [8]. This method recursively divides the image into subregions at increasingly fine resolutions. We adopt this idea in our NN search by limiting each feature point in the image to find its NN only in the same subregion from a candidate class at each level. So the feature searching set in the candidate class is reduced from the whole image (top level, or level 0) to only the corresponding subregion (finer level), see Figure 2 for details. This spatial restriction enhances the robustness of NN search by reducing the effect of noise due to wrong matches from other subregions. Then the learned distances from all levels are merged together as pyramid combination.

In addition, we find in our experiments that a single level spatial restriction at a finer resolution makes better recognition accuracy compared to the top level especially for those images with geometric scene structure, although the accuracy is slightly lower than the pyramid combination of all levels. Since the candidate searching set is smaller in a finer level, which requires less computation cost for the NN search, we can use just a single level spatial restriction of the

learned I2C distance to speed up the classification for test images. Compared to the top level, a finer level spatial restriction not only reduces the computation cost, but also improves the recognition accuracy in most datasets. For some images without geometric scene structure, this single level can still preserve the recognition performance due to sufficient features in the candidate class.

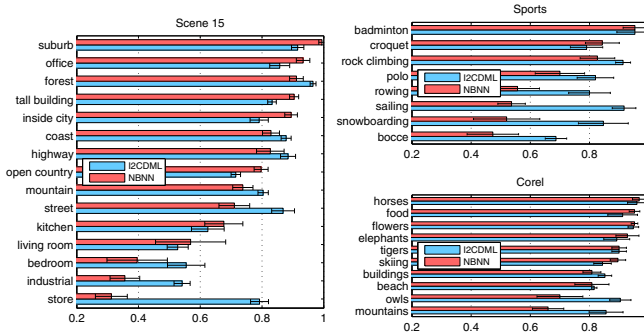
We also use the method of learning I2C distance function proposed in [16] to combine with the learned Mahalanobis I2C distance. The idea of learning local distance function is originally proposed by Frome *et al.* and used for image classification and retrieval in [6,5]. Their method learns a weighted distance function for measuring I2I distance, which is achieved by also using a large margin framework to learn the weight associated with each local feature. Wang *et al.* [16] have used this idea to learn a weighted I2C distance function from each image to a candidate class, and we find our distance metric learning method can be combined with this distance function learning approach. For each class, its weighted I2C distance is multiplied with our learned Per-Class matrix to generate a more discriminative weighted Mahalanobis I2C distance. Details of this local distance function for learning weight can be found in [6,16].

## 3 Experiment

### 3.1 Datasets and Setup

We evaluate our proposed method on four popular datasets: Scene-15, Sports, Corel and Caltech 101 dataset. We describe them briefly as follows:

- **Scene-15.** Scene dataset consists of 15 scene categories, among which 8 were originally collected by Oliva *et al.* [15], 5 added by Li *et al.* [4] and 2 from Lazebnik *et al.* [9]. Each class has about 200 to 400 images, and the average image size is around  $300 \times 250$  pixels. Following [9], we randomly select 100 images per class for training and test on the rest. The mean per-class recognition rate is reported as accuracy.
- **Sports.** Sports event dataset is firstly introduced in [10], consisting of 8 sports event categories. The number of images in each class ranges from 137 to 250, so we follow [10] to select 70 and 60 images per class for training and test respectively. Since images in this dataset are usually very large, they are first resized such that the largest x/y dimension is 500.
- **Corel.** Corel dataset contains 10 scene categories published from Corel Corporation. Each class contains 100 images, and we follow [14] to separate them randomly into two subsets of equal size to form the training and test set. All the images are of the size  $384 \times 256$  or  $256 \times 384$ .
- **Caltech 101.** Caltech 101 dataset is a large scale dataset containing 101 categories [3]. The number of images in each class varies from about 30 to 800. This dataset is more challenging due to the large number of classes and intra-class variances. Following the widely used measurement by the community we randomly select 15 images per class for training. For test set, we also select 15 images from each class and report the mean accuracy.



**Fig. 3.** Per-category recognition accuracy for comparison of I2CDML with NBNN

Since the training and test set are selected randomly, we repeat the experiment for 5 times in each dataset and report the average result. For feature extraction, we use dense sampling strategy and SIFT features [12] as our descriptor, which are computed on a  $16 \times 16$  patches over a grid with spacing of 8 pixels for all datasets. This is a simplified method compared to some papers that use densely sampled and multi-scale patches to extract large number of features, which helps in the performance results but increases the computational complexity. We name our method as I2CDML, short for Image-To-Class distance metric learning.

### 3.2 Results on Scene-15, Sports and Corel Datasets

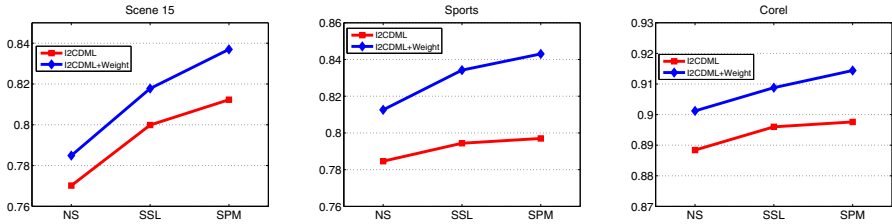
We first compare our proposed I2CDML method with NBNN [1] on Scene-15, Sports and Corel datasets to evaluate our learned metrics. Table 1 shows the recognition accuracy averaged of all classes for the three datasets. We can see that our method significantly outperforms NBNN in every dataset, especially in Sports dataset where the improvement is above 10%. Then we investigate the details by comparing the classification accuracy for each class in Figure 3. For those easily classified categories, our method is comparable to NBNN. Moreover, for those challenging categories that NBNN performs poorly (for example the worst three categories in Scene-15, the worst four in Sports, and the worst two in Corel, as indicated in Figure 3), our method can improve the accuracy substantially. Therefore our method improves the average accuracy by emphasizing the classification on challenging categories and yet maintains the performance for the easily classified categories.

**Table 1.** Comparing I2CDML to NBNN for recognition accuracy (%)

Method	Scene-15	Sports	Corel
<b>I2CDML</b>	<b>77.0 ± 0.60</b>	<b>78.5 ± 1.63</b>	<b>88.8 ± 0.93</b>
NBNN [1]	72.3 ± 0.93	67.6 ± 1.10	85.7 ± 1.20

**Table 2.** Comparing I2CDML to its integration for recognition accuracy (%)

Method	Scene-15	Sports	Corel
I2CDML	$77.0 \pm 0.60$	$78.5 \pm 1.63$	$88.8 \pm 0.93$
I2CDML+SPM	$81.2 \pm 0.52$	$79.7 \pm 1.83$	$89.8 \pm 1.16$
I2CDML+Weight	$78.5 \pm 0.74$	$81.3 \pm 1.46$	$90.1 \pm 0.94$
<b>I2CDML+ SPM+Weight</b>	<b><math>83.7 \pm 0.49</math></b>	<b><math>84.3 \pm 1.52</math></b>	<b><math>91.4 \pm 0.88</math></b>



**Fig. 4.** Comparing the performance of no spatial restriction (NS), spatial single level restriction (SSL) and spatial pyramid match (SPM) for both I2CDML and I2CDML+Weight in all the three datasets. With only spatial single level, it achieves better performance than without spatial restriction, although slightly lower than spatial pyramid combination of multiple levels. But it requires much less computation cost for feature NN search.

Then we show in Table 2 the improved I2C distance through spatial pyramid restriction from the idea of spatial pyramid match in [9] and learning weight associated with each local feature in [16]. Both strategies are able to augment the classification accuracy for every dataset, and we find that SPM provides additional improvement than learning weight in Scene-15 dataset but less improvement in the other two datasets. This is likely due to the geometric structure of Scene-15 that matches with the spatial equally divided subregions very well, while in the other two datasets discriminative local features for generating the weighted I2C distance have a more important role for classification. Nevertheless, by using both strategies we get the best results in all the three datasets.

Since a spatial single level at a finer resolution will reduce the computation cost required for feature NN search, we also compare its performance with spatial pyramid combining multiple levels as well as the original size without using spatial restriction. As shown in Figure 4, this spatial single level is able to improve the accuracy compared to no spatial restriction, not only on scene constraint datasets (Scene-15 and Corel) but also on Sports event dataset that does not have geometric scene structure. Though the performance is slightly lower than the pyramid combining all levels, it saves the computation cost for both feature NN search and distance metric learning. So this spatial single level strategy will be very useful for improving the efficiency.



**Table 3.** Comparing to recently published results for recognition accuracy (%)

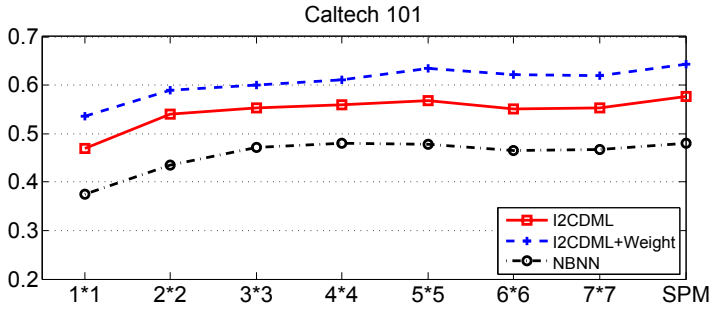
Scene-15		Sports		Corel	
Ours	<b>83.7</b>	Ours	<b>84.3</b>	Ours	<b>91.4</b>
Lazebnik <i>et al.</i> [9]	81.4	Li <i>et al.</i> [10]	73.4	Lu <i>et al.</i> [13]	77.9
Liu <i>et al.</i> [11]	83.3	Wu <i>et al.</i> [21]	78.5	Lu <i>et al.</i> [14]	90.0
Bosch <i>et al.</i> [2]	83.7	Wu <i>et al.</i> [22]	84.2		
Yang <i>et al.</i> [24]	80.3				
Wu <i>et al.</i> [22]	84.1				

We compare our best results with recently published result for every dataset. All the results are listed in Table 3. In Scene-15 dataset, many researchers reported their recent results and most of them also incorporate SPM to improve the accuracy. Lazebnik *et al.* [9] first proposed SPM and combined with the Bag-of-Word (BoW) strategy, achieving 81.4%. Bosch *et al.* [2] also incorporated SPM in pLSA to achieve 83.7%. The best result so far as we know is 84.1% by Wu *et al.* [22], who replace Euclidean distance by histogram intersection in BoW combined with SPM. Although our result is slightly lower than their results, we notice they have used multi-scale and denser grid to extract feature as well as combining additional Sobel gradient, while our feature extraction is very simple but still comparable to theirs. When using the same configuration, their approach is worse than ours, as either indicated in [22] as well as implemented by us using their published LibHIK<sup>1</sup> code, which is  $81.36 \pm 0.54$  using CENTRIST [23] and  $78.66 \pm 0.44$  using SIFT [12] in our implementation. For Sports dataset, Li *et al.* [10] who published them reported 73.4%, and Wu *et al.* improved it to 78.5% and 84.2% in [21] and [22] respectively, where the later one used the same configuration as their experiment in Scene-15. Nevertheless, our result is still comparable to theirs. For Corel dataset, our result is again better than the previous published results [14][13] even without using color information. All these results show that our method can achieve state-of-the-art performance on these datasets using relatively small feature set.

### 3.3 Results on Caltech 101 Dataset

We also evaluate our method on Caltech 101 to illustrate its scalability on datasets with more classes. In our experiment we only select 15 images per class for training, same as most previous studies. In [1], they extracted SIFT features using multi-scale patches densely sampled from each image, which result in much redundant features on the training set (about 15000 to 20000 features per image). So the NN search for I2C distance calculation takes expensive computation cost. Even using KD-Tree for acceleration, it takes about 1.6 seconds per class for the NN search of each test image [1] and thus around 160 seconds for 101 classes to classify only one test image. This is unacceptable during the testing phase and makes it difficult for real-world application. In our experiment, we only generate

<sup>1</sup> <http://www.cc.gatech.edu/cpl/projects/libHIK/>



**Fig. 5.** Comparing the performances of I2CDML, I2CDML+Weight and NBNN from spatial division of  $1 \times$  to  $7 \times 7$  and spatial pyramid combination (SPM) on Caltech 101.

less than 1000 features per image on average using our feature extraction strategy, which are about  $1/20$  compared to the size of feature set in [1]. We also use single level spatial restriction to constrain the NN search for acceleration. For each image, we divide it from  $2 \times 2$  to  $7 \times 7$  subregions and test the performance of I2CDML, NBNN and I2CDML+Weight. Experiment results of without using spatial restriction ( $1 \times 1$  region) as well as spatial pyramid combining all levels is also reported.

From Figure 5, we can see that without using spatial restriction, the performance of NBNN is surprisingly bad. The reason that NBNN performs excellently as reported in [1] using complex features while poorly in our small feature set implies that the performance of NBNN relies heavily on the large redundant of training feature set, which needs expensive computation cost for the NN search. For comparison, our I2CDML augments the performance of I2C distance significantly, while combining the learned weight further improves the performance. Compared to the other three datasets, this dataset contains more classes and less training images per class, which makes it more challenging. So the large improvement over NBNN indicates that our learning procedure plays an important role to maintain an excellent performance using small number of features under such challenging situation, which also requires much less computation cost in the testing phase.

From  $2 \times 2$  to  $7 \times 7$  subregions of spatial restriction, due to the regular geometric object layout structure of images in this dataset, the performance is further improved for all methods compared to without using spatial restriction. Though the results on spatial division from  $3 \times 3$  to  $7 \times 7$  do not change much, the computation cost for NN search continues decreasing with finer spatial division. For  $7 \times 7$  spatial division, the size of feature set in the candidate class is  $1/49$  of the original image without using spatial restriction. The best result on single spatial restriction is 63.4% by I2CDML+Weight on  $5 \times 5$  spatial division, which is close to the result of spatial pyramid combining all levels (64.4%) but is more efficient. NBNN can also benefit from this spatial restriction, but its result is still unacceptable for classification task using such small feature set.

Our best result is also comparable to the best reported result of NBNN (65%) in [1], which uses large number of densely sampled multi-scale local features as well as pixel location information for their I2C distances to achieve such state-of-the-art performance. The size of candidate feature set they used is about 20 times more than ours using the whole image and nearly 1000 times compared our spatial restriction of  $7 \times 7$  subregions. So our implementation needs much less computation cost for the NN search during the on-line testing phase with the additional off-line training phase, whilst the result is comparable to theirs. Although we cannot reproduce their reported result in our implementation, we believe our comparison should be fair as we use the same feature set for all methods and the experiment has shown that our method achieves significant improvement on such large-scale dataset with much efficient implementation.

## 4 Conclusion

Image-To-Class distance relies heavily on the large number of local features in the training set and test images, which need heavy computation cost for the NN search in the testing phase. However, using small number of features results in poor performance. In this paper, we tried to improve the performance of this distance and speed up the testing phase. We added a training phase by proposing a distance metric learning method to learn the I2C distance. A large margin framework has been formulated to learn the Per-Class Mahalanobis distance metrics, with a gradient descent method to efficiently solve this optimization problem. We also discussed the method of enhancing the discrimination of the learned I2C distance for performance improvement. These efforts made the I2C distance perform excellent even using small feature set. For further accelerating the NN search in the testing phase, we adopted single level spatial restriction, which can speed up the NN search significantly while preserving the classification accuracy. The experiment results on four datasets of Scene-15, Sports, Corel and Caltech 101 verified that our I2CDML method can significantly outperform the original NBNN especially for those challenging categories, and such I2C distance achieved state-of-the-art performance on most datasets.

## References

1. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
2. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. TPAMI (2008)
3. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. CVPR Workshop on Generative-Model Based Vision (2004)
4. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
5. Frome, A., Singer, Y., Malik, J.: Image retrieval and classification using local distance functions. In: NIPS (2006)

6. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
7. van Gemert, J.C., Geusebroek, J.M., Veenman, C.J.: Kernel codebooks for scene categorization. In: ECCV (2008)
8. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV (2005)
9. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
10. Li, J., Fei-Fei, L.: What, where and who? Classifying events by scene and object recognition. In: ICCV (2007)
11. Liu, J., Shah, M.: Scene modeling using co-clustering. In: ICCV (2007)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2) (2004)
13. Lu, Z., Ip, H.H.: Image categorization by learning with context and consistency. In: CVPR (2009)
14. Lu, Z., Ip, H.H.: Image categorization with spatial mismatch kernels. In: CVPR (2009)
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42(3) (2001)
16. Wang, Z., Hu, Y., Chia, L.T.: Learning instance-to-class distance for human action recognition. In: ICIP (2009)
17. Rasiwasia, N., Vasconcelos, N.: Scene classification with low-dimensional semantic spaces and weak supervision. In: CVPR (2008)
18. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2005)
19. Weinberger, K.Q., Saul, L.K.: Fast solvers and efficient implementations for distance metric learning. In: ICML (2008)
20. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (2009)
21. Wu, J., Rehg, J.M.: Where am I: Place instance and category recognition using spatial PACT. In: CVPR (2008)
22. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2009)
23. Wu, J., Rehg, J.M.: CENTRIST: A visual descriptor for scene categorization. Technical Report GIT-GVU-09-05, GVU Center, Georgia Institute of Technology (2009)
24. Yang, J., Lu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)

# Extracting Structures in Image Collections for Object Recognition\*

Sandra Ebert<sup>1,2,\*\*</sup>, Diane Larlus<sup>1,\*\*</sup>, and Bernt Schiele<sup>1,2</sup>

<sup>1</sup> Department of Computer Science, TU Darmstadt, Germany

<sup>2</sup> MPI Informatics, Saarbrücken, Germany

**Abstract.** Many computer vision methods rely on annotated image sets without taking advantage of the increasing number of unlabeled images available. This paper explores an alternative approach involving unsupervised structure discovery and semi-supervised learning (SSL) in image collections. Focusing on object classes, the first part of the paper contributes with an extensive evaluation of state-of-the-art image representations. Thus, it underlines the decisive influence of the local neighborhood structure and its direct consequences on SSL results and the importance of developing powerful object representations. In a second part, we propose and explore promising directions to improve results by looking at the local topology between images and feature combination strategies.

**Keywords:** object recognition, semi-supervised learning.

## 1 Introduction

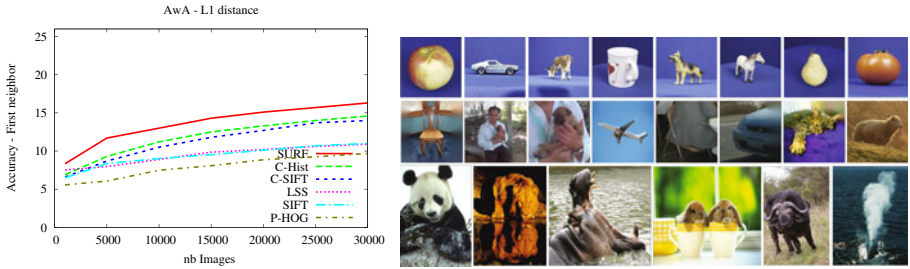
Supervised learning is the de facto standard for many computer vision tasks such as object recognition or scene categorization. Powerful classifiers can obtain impressive results but require a sufficient amount of annotated training data. However, supervised methods have important limitations: Annotation is expensive, prone to error, often biased, and does not scale. Obtaining the required training data representing all relevant aspects of a given category is difficult but key to success for supervised methods. Facing these limitations we argue that the computer vision community should move beyond supervised methods and more seriously tap into the vast collections of images available today.

In particular, we look at the *local structure* of the data (links between images here) in an *unsupervised way*. For larger datasets, this local neighborhood becomes more reliable: Two semantically similar images (belonging to the same class) have a higher probability to be also similar in image representation space for increasing database sizes (see Fig. [II](#)). *Semi-supervised learning* (SSL), the second direction explored here, uses such local neighborhood relations and few labeled images to predict the label of new images. The local structure has in both cases a strong influence on the overall performance of such approaches.

---

\* This work was supported by a Google Research Award.

\*\* The first two authors contributed equally. Names are ordered alphabetically.



**Fig. 1.** (Left) For each image, we look at the most similar image (L1 distance NN) to see how often these couples belong to the same class. This increases for larger sets. (Right) Top: ETH, middle: our Cropped PASCAL, bottom: AWA datasets.

This paper is organized in two parts. First, we contribute a study of different representations and SSL algorithms on three image collections of increasing size and difficulty for image categorization (Sec. 3). We show that the results depend on the neighborhood structure induced by object representations and on the graph structure parameters rather than on the particular SSL algorithm employed. We also show that results obtained on the local neighborhood directly transfer to SSL results.

Motivated by these conclusions, the second part of the paper presents different ways of improving the connections between images in the local neighborhood structure. Among the considered strategies: the topology of the dataset is used to refine the existing connections (Sec. 4), and different features are combined (Sec. 5). Results show improvements for both the structure and the SSL predictions on all datasets.

**Related Work.** The use of large image collections is obviously not a novel idea. [1] directly discovers image clusters, while other approaches aim to globally partition the database in image sets sharing more general concepts [2]. Multi instance learning methods deal with weak or incomplete annotations [3]. Some methods use the web as an external source of information to get many but noisy annotations [4]. Active learning methods aim to identify missing annotations [5]. Finally, attempts are made to make the annotation process more appealing [6]. None of this prior work however systematically analyzes the suitability of today’s image and object representations for unsupervised local structure extraction.

Semi-supervised learning (SSL) has been applied to several computer vision problems. Partial labeling of pixels is used as an input for segmentation [7]. Image level annotations are used to find object parts [8]. But only a couple of methods apply SSL to predict labels at the image level from a few annotated images. Of particular interest are [9] using random forests and [10] using boosting, both in an SSL framework. Closer to our work, [11] focuses on graph based propagation algorithms and proposes efficient approximations to scale SSL methods to large datasets. In machine learning, SSL methods have been used with success

for many tasks (e.g. digit recognition, text classification, or speech recognition, see [12] for a survey). Among SSL, graph-based methods play an important role as they concentrate on the local structure of data [13,14]. Most approaches however focus on SSL algorithms rather than on the underlying structure. In this work, we analyze the local neighborhood in detail to improve the performance of SSL graph-based algorithms for image data.

There are very few studies that compare SSL methods on images. [12] contains a single small image set and [15] considers digits and faces only. In both cases, representations are different from commonly used image descriptors for recognition. Therefore, this paper focuses on the important problem of how well standard representations are suited for unsupervised structure discovery as well as SSL and how the structure can be improved such that also SSL can benefit.

## 2 Datasets and Image Representations

We consider three datasets with increasing number of object classes, number of images, and difficulty. Some of the images are shown in Fig. 1.

*ETH-80* (ETH) [16] contains 3,280 images divided in 8 object classes and 10 instances per class. Each instance is photographed from 41 viewpoints in front of a uniform background. This controlled dataset ensures that a strong local structure exists between images making it a perfect toy dataset for our task.

*Cropped PASCAL* (C-PASCAL) is based on the PASCAL VOC challenge 2008 training set [17]. Bounding box (BB) annotations are used to extract the objects. Consequently semantic connections between images and SSL predictions on this new dataset can be evaluated in our multi-class protocol. To discard information contained in the aspect ratio of the BB, squared regions (rescaled to 102x102 pixels) are extracted using the larger side of the BB and objects smaller than 50 pixels are discarded. To avoid that a class dominates the evaluation, we subsampled the largest class ‘people’ from 40% to 16% (the 2nd largest class being ‘chair’ with ~11%). The set contains 6,175 images of aligned objects from 20 classes but with varying object poses, challenging appearances, and backgrounds.

*Animal with Attributes* (AWA) [18] is a large and realistic dataset with 30,475 images and 50 classes, without alignment. Objects are located anywhere in the image, in difficult conditions and poses, which complicates the task of finding images containing similar object classes. While being the most challenging dataset in this evaluation, it is the kind of data we are eventually aiming for.

**Representations.** This paper uses a large spectrum of representations employed by state-of-the art recognition methods [17]. For the first two datasets we consider 7 complementary descriptors: 3 global descriptors (HOG [19], Gist [20], pyramid bag-of-features (P-BoF) [21]), 3 bag-of-features representations (BoF) with different detectors and descriptors, and a texture descriptor (TPLBP [22]). Our HOG implementation uses 9-bins histograms of gradient orientations, locally normalized over contrast, extracted using a dense grid of non-overlapping cells (8x8 pixels). For the Gist scene descriptor we use the code of [20]. P-BoF

features are computed with the implementation of [21]. It extracts patches on 4 different levels and a visual vocabulary of 200 words. Concurrently, we extract bag-of-features representations. We combine Harris (Har-BoF) or Hessian-Affine (Hess-BoF) detectors [23], with SIFT [24] and build visual vocabularies of 10,000 words. We also use C-SIFT based on the code of [25] (Color-SIFT descriptors for Harris points, 2,000 words vocabulary). Finally, the local texture descriptor called Three Patch Local Binary Pattern (TPLBP) [22] considers 3 neighboring patches of size  $3 \times 3$  arranged in a circle a single bit value for each pixel. For AWA, we use 7 descriptors: the 6 publicly available features [18] (color histograms (C-Hist), Local-Self-Similarity (LSS), Pyramid HOG (P-HOG), bag-of-features representations involving SIFT, color-SIFT (C-SIFT), and SURF descriptors) and the Gist descriptor that we computed additionally.

### 3 Local Structure and SSL Study

As stated before, this paper looks at two related tasks: *local structure extraction* and the use of this structure for *semi-supervised learning* (SSL). We focus on the question whether today's object class representations are suitable for local structure discovery and how well these observations transfer to SSL.

The following first analyzes neighborhood structures and then compares four different graph-based algorithms for SSL.

**Local structure discovery.** For all three datasets, we analyze neighborhood structures of different object representations, for the L1 and L2 distance measures<sup>1</sup>. We focus on  $k$ -nearest neighbors ( $k$ -NN) structures which have better connectivity and lead to more intuitive structures than e.g.,  $\epsilon$ -neighborhood graphs [26]. These properties are also important for SSL algorithms.

**Experiments.** To evaluate the quality of the  $k$ -NN structure for an image, we calculate the percentage of neighbors belonging to the same class as this image. Averaging this percentage over all images results in the overall  $k$ -NN structure accuracy. Intuitively, this evaluates how often the  $k$ -NN structure connects images from the same class, and how much semantic information it contains.

The left side of Tab. 1 shows L1 and L2 performances for the nearest (1-NN) and the 10 nearest neighbors (10-NN), for all three datasets<sup>2</sup>. First, we see that L1 constantly outperforms L2 for all representations and all datasets.

Also, we observe that results significantly differ between the different representations. Global descriptors like P-BoF or Gist work well for ETH and C-PASCAL as objects are mostly aligned in those databases. Local descriptors are better suited for the more challenging AWA dataset.

Finally, the 1-NN and 10-NN exhibit different behaviors. Some features are more robust for larger numbers of neighbors. For instance for C-PASCAL, Hess-BoF is the third best descriptor when looking at 1-NN structures, with 31.3%,

<sup>1</sup> The  $\chi^2$  measure was considered but not reported as it gave similar results as L1.

<sup>2</sup> In all tables for both NN and SSL, best representation per configuration: gray cell (max per column); best configuration: bold numbers (max per line); overall best: red.



**Table 1.** Quality of the nearest neighbor and the 10 nearest neighbors on the left part, transductive propagation results on the right part, for L1, L2 (Sec. 3) and L2-Context (Sec. 4.1). Only the 3 best descriptors are shown for ETH.

	Features	NN quality						SSL results					
		L1		L2		L2-ctxt		L1		L2		L2-ctxt	
		k=1	k=10	k=1	k=10	k=1	k=10	acc	var	acc	var	acc	var
ETH	C-SIFT	<b>96.6</b>	89.0	80.5	63.1	92.8	82.7	<b>89.0</b>	0.6	60.9	2.0	83.7	1.4
	Gist	<b>93.6</b>	<b>85.4</b>	92.9	83.5	<b>93.6</b>	84.8	83.1	1.4	82.5	1.0	<b>84.5</b>	0.8
	HOG	<b>96.9</b>	<b>88.6</b>	95.5	86.2	<b>96.9</b>	<b>88.6</b>	84.5	1.8	83.3	1.7	<b>86.8</b>	1.3
C-PASCAL	C-SIFT	<b>32.6</b>	<b>19.6</b>	24.2	13.9	30.1	17.8	<b>24.0</b>	0.4	16.6	2.2	20.5	0.4
	Gist	30.8	24.3	29.5	23.5	<b>31.8</b>	<b>24.9</b>	<b>28.4</b>	0.3	27.5	0.4	28.1	0.8
	HOG	27.3	21.4	22.0	17.3	<b>34.6</b>	<b>26.8</b>	19.2	2.3	13.9	2.4	<b>28.8</b>	1.6
	Har-BoF	<b>28.7</b>	<b>16.2</b>	17.8	10.4	25.1	13.6	<b>20.1</b>	0.5	13.1	2.7	15.7	0.5
	Hess-BoF	<b>31.3</b>	<b>17.9</b>	20.1	11.2	26.7	15.2	<b>21.6</b>	0.7	15.3	2.3	16.5	1.1
	P-BoF	<b>28.5</b>	<b>22.3</b>	24.1	17.7	24.1	17.7	<b>28.4</b>	0.9	20.2	1.2	20.6	0.9
AWA	TPLBP	<b>33.5</b>	<b>26.2</b>	26.9	20.4	26.9	20.4	<b>29.5</b>	0.9	20.4	2.0	20.5	1.9
	C-Hist	<b>14.5</b>	<b>9.2</b>	9.8	6.6	12.2	8.3	8.4	0.2	5.7	0.1	<b>8.5</b>	0.2
	C-SIFT	14.2	9.2	12.2	8.0	<b>15.4</b>	<b>10.4</b>	8.0	0.2	7.0	0.1	<b>10.3</b>	0.2
	Gist	12.1	8.1	12.0	8.1	<b>15.0</b>	<b>10.3</b>	7.2	0.2	7.4	0.2	<b>10.9</b>	0.1
	LSS	10.9	7.8	8.3	6.3	<b>11.7</b>	<b>8.1</b>	6.9	0.1	5.5	0.3	<b>8.2</b>	0.3
	PHOG	<b>9.7</b>	6.7	7.7	5.6	8.9	<b>7.0</b>	6.3	0.2	5.4	0.1	<b>7.0</b>	0.1
	SIFT	11.0	8.1	10.4	7.6	<b>12.4</b>	<b>8.8</b>	7.7	0.2	7.3	0.3	<b>9.2</b>	0.2
SURF	<b>16.4</b>	10.6	11.7	8.0	14.3	<b>10.7</b>	9.0	0.1	6.8	0.3	<b>10.4</b>	0.2	

but loses almost half of the performance when considering 10-NN (17.9%). On the contrary, P-BoF gives poor results for 1-NN but is more robust for 10-NN (22.3%). When considering SSL results, we will refer mainly to the 10-NN structures, as graphs are using  $k$ -NN structures with large enough values of  $k$ .

**Semi-supervised learning.** We use the previously studied  $k$ -NN structure and few labels in a graph and analyze several SSL methods for the object recognition problem. These methods build a graph  $(X, Y)$  where the nodes  $X = \{X_l, X_u\}$  represent images and  $Y = \{Y_l, Y_u\}$  are the labels.  $(X_l, Y_l)$  are labeled images and  $(X_u, Y_u)$  are unlabeled images. A graph is represented by an adjacency matrix  $W$  built from the  $k$ -NN structure. The degree of each node is  $d_{ii} \leftarrow \sum_j w_{ij}$  and defines the diagonal matrix  $D$ . Here, we evaluate *non-symmetric* (directed) graphs. We do not evaluate fully connected graphs due to their computational complexity and memory requirements. We also considered weighted graphs but found that performance did not improve significantly.

Graph-based methods distribute labels from labeled to unlabeled nodes. In our experiments, we compare four methods covering a broad range of possible strategies. These methods are designed for binary problems, and expandable to multi-class problems with  $n$  classes, by splitting them into  $n$  one-versus-all binary problems, that share the same graph structure. All algorithms follow the same pattern. First, labels are initialized, with  $Y_l$  taking values in  $\{1, -1\}$  and elements of  $Y_u$  set to 0 resulting in  $\hat{Y}^{(0)}$ . Then labels are updated iteratively  $\hat{Y}^{(t+1)} \leftarrow L\hat{Y}^{(t)}$  for a certain number of iterations<sup>3</sup>. This part differs for each method, and is briefly described below.

<sup>3</sup> Typically a small number of iterations is used to avoid over-fitting.

*Gaussian Fields Harmonic Functions (GFHF)* [14] uses a transition probability matrix  $L = D^{-1}W$  to propagate the labels. Original labels cannot change.

*Quadratic Criterion (QC)* [27] is a variant of the previous method allowing the original labels to change, which can help for ambiguous representations. It also introduces a regularization term for better numerical stability.

*Local Global Consistency (LGC)* [13] uses a normalized graph Laplacian  $L = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$  instead of transition probabilities. The initial labels are also allowed to change, but in a regularized way. A parameter  $\alpha$  (set to 0.5) regularizes the modifications to limit overwritten labels and weights how much newly predicted labels are trusted compared to original ones,  $\hat{Y}^{(t+1)} \leftarrow \alpha L\hat{Y}^{(t)} + (1 - \alpha)\hat{Y}^{(0)}$ .

*Discrete Regularization (DR)* [28] incorporates local graph properties by looking at the degree of two neighboring nodes. An additional cost function reduces the influence of nodes with many connections.

**Experiments.** We apply these algorithms to all datasets and focus on the following aspects: the differences between the 4 SSL algorithms, between the different representations, and the influence of the local structure on SSL results.

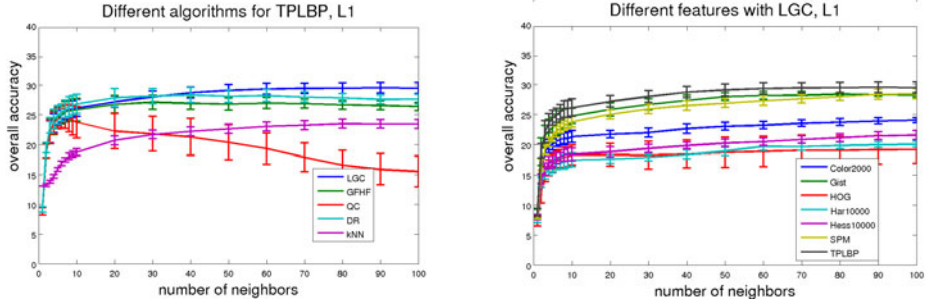
We evaluate transductive results (i.e. prediction for the remaining labels) with 10% labeled data, for all datasets on their different representations for L1 and L2. All experiments randomly select 5 sets of labeled data, and produce mean and variance of the overall multi-class accuracy on unlabeled data only (transductive results). For comparison, we also use the supervised k-NN classifier, based on the same representations and measures. Some representative results<sup>2</sup>, which illustrate our main findings, are summarized in Fig. 2 and in Tab. 1 (right).

i) *Graph structure and different algorithms.* The first experiment varies the number of neighbors  $k$  in the graph, for the different algorithms. Fig. 2 (left) shows the obtained performances for TPLBP, which performed best for C-PASCAL in the case of local structures. As we can see, the number of neighbors  $k$  is a crucial parameter and an optimal value exists. This value appeared to be dataset- and SSL algorithm dependent. A minimum number is required to perform reasonably. Too small  $k$  values result in a graph with disconnected components, where no information is propagated and some images are not classified.

Once the correct parameters for the graphs are chosen, there are surprisingly small differences between SSL methods. For instance, L1 numbers vary between 29.5% for LGC and 24.6% for QC. This emphasizes our claim that the structure is more important than the algorithms. LGC is more stable across experiments and the QC method tends to achieve lower results. This algorithm allows to change the original labels but has no regularization parameter like LGC, leading to many changes in the original labels, and accuracy drops for large  $k$  values. Finally, all SSL results outperform the best k-NN result (k=80) of 23.5%, showing the benefits of the unlabeled data in the classification process.

In the remainder, we use LGC [13], as it exhibited stable results across multiple settings, and its best settings determined from our parameter study.

ii) *Image representations.* As before in the NN study, we notice large differences between image representations (Fig. 2 (right) for C-PASCAL and Tab. 1 for ETH



**Fig. 2.** For C-PASCAL. Mean/Variance of overall accuracy on unlabeled data only for LGC, GFHF, QC, and DR and k-NN (left). Different features with LGC (right).

and AWA). For C-PASCAL<sup>4</sup>, accuracies vary from 19.2% for HOG to 29.5% for TPLBP and for AWA from 6.3% for P-HOG to 9% for SURF, with L1. Second, we observe the existing gap between the representations and the task. Our toy dataset ETH exhibits good values, meaning that for datasets with an obvious underlying structure (same objects and views), it is accurately extracted and used by the propagation algorithm. We were able to obtain satisfying results even with minimal supervision. For the more challenging C-PASCAL and AWA datasets, numbers are more disappointing. We can conclude that today’s image representations are still not rich enough for building good semantic structures.

iii) *Transfer from neighborhood structure.* Tab. 1 shows that the results<sup>2</sup> of the 10-NN structures (left side) transfers directly to the SSL performances (right side) for each dataset, including the observed semantic gap. 10-NN performance is more consistent with the SSL algorithms as the latter need a minimum number of connections to propagate the labels. Note that 1-NN structure quality is always higher than SSL results because it gives only an intuition on the probability for an image to transfer the correct label to its first neighbor. We could reach this number for about 50% of the images labeled.

**Summary.** In this section, we studied the local neighborhood structure and its influence on different SSL algorithms. We observed that the parameters that determine the local neighborhood structure (image representations, value of  $k$ , etc.) result in larger differences in performance than the particular choice of the SSL algorithm. ETH presents high quality neighbors and the semantic structure of the dataset is captured accurately. For more realistic datasets, like C-PASCAL and AWA, the quality of neighbors is disappointing and underline the existing gap between considered categories’ appearance and today’s computer vision representation. This limitation also transfers to SSL results.

<sup>4</sup> Note that the non-balanced C-PASCAL (dominated by the well recognized person class) shares similar observations with our C-PASCAL, across all experiments, with higher overall numbers. For instance here, L1 varies from 27.4% for HOG and 47.1% for TPLBP.

## 4 Improving the Local Structure between Images

The previous section showed that local neighborhood structures capture some semantic information between images, but still a gap exists between the connections in a structure and the object categories, leaving ample room for improvement. Also, we showed that this structure has a stronger influence on the results than the SSL method itself. Therefore, this and the following section consider different directions to improve the quality of the connections between images. We want to improve the local structure without any learning involved, trying again to move away from supervised methods. Our goal is to build an improved *unsupervised local neighborhood structure*, which consequently can be generic, and does not depend on the considered SSL problem.

In the following we explore to which extent the neighborhood structure can be improved without labels, using only topological information of the dataset itself and look at its influence on (i) the local structure itself, and (ii) the SSL results. Sec. 4.1 considers context measures as an improvement over standard measures and Sec. 4.2 shows the benefit of symmetric relations between neighbors.

### 4.1 Context Measures

We consider the contextual measure, proposed in [29] for the image retrieval task. This context measure is applied for L2<sup>3</sup> to our problem.

**Principle.** When trying to decide if two images are close, the answer is often given for a given context. We do not only look at the images themselves, but also at the surrounding images. This is the intuition behind the contextual measure [29], that computes the distance from a first descriptor  $p$  to another descriptor  $q$  in the context of  $u$  using:  $L2_{ctx}(q, p|u) = \operatorname{argmin}_{0 \leq \omega \leq 1} \{ \|q - (\omega p + (1 - \omega)u)\|_2 \}$ . The context vector  $u$  is obtained by computing the mean vector of the  $l$  nearest neighbors ( $l=100$  in our experiments) of  $p$  in the collection.

**Experiments.** Tab. 1 summarizes the results<sup>2</sup> obtained for the L2-context measure, in comparison with the L1 and L2 measures considered in our previous study. From this table we can make the following observations. First, context measure yields a consistent improvement to the L2 measure. For 1-NN, this improvement represents about 9% on average for ETH, almost 5% on average for C-PASCAL and 2.5% for AWA. The same applies for the SSL results: we note 11% improvement for ETH, about 3% for C-PASCAL and for AWA, on average. Sparse vectors (e.g. Hess-BoF or Har-BoF) benefit the most. Again we observe (cf. Tab. 1) the consistency between the NN quality and the corresponding SSL results, already underlined in the previous section's study.

Interestingly, L2-ctx brings L2 to the level of L1 and sometimes outperforms it. Context measures are a promising direction and one could expect further improvement from the context version of L1. As no closed-form solution exists for L1, this new measure will be difficult to scale to very large datasets. Therefore, we consider a different strategy which scales more easily in the following.

<sup>5</sup> A closed-form solution is available for L2, making computations faster.

**Table 2.** Quality of the nearest and the 10 nearest neighbors, chosen with distance- or a rank-based strategy, for AWA

Features		1-NN						10-NN					
		Dist			Rank			Dist			Rank		
		L1	L2	L2-cxt	L1	L2	L2-cxt	L1	L2	L2-cxt	L1	L2	L2-cxt
AWA	C-Hist	14.5	9.8	12.1	<b>16.8</b>	12.1	12.9	9.2	6.6	8.3	10.6	7.9	8.5
	C-SIFT	14.1	12.1	15.3	<b>19.1</b>	14.8	16.6	9.1	8.0	10.4	11.6	9.4	10.6
	Gist	12.1	11.9	15.0	14.8	14.4	<b>15.2</b>	8.0	8.0	10.2	9.7	9.7	10.2
	LSS	10.8	8.2	11.7	<b>14.6</b>	10.4	12.3	7.8	6.2	8.0	9.5	7.1	8.1
	PHOG	9.7	7.6	8.8	<b>12.2</b>	9.3	9.8	6.7	5.6	6.9	7.9	6.2	7.0
	SIFT	11.0	10.3	12.3	<b>12.5</b>	11.5	<b>12.5</b>	8.0	7.6	8.8	8.9	8.4	8.9
NN quality	SURF	16.3	11.7	14.3	<b>23.0</b>	14.2	15.8	10.5	8.0	10.7	14.2	8.9	10.8

**Table 3.** Left: non-symmetric graphs on rank-based structures. Middle: symmetric graphs on distance-based structures. Right: symmetric graph and rank based structures. The improvement obtained in comparison to Tab. 1 is shown in the gain column.

Feat.		ETH - SSL results																	
		Rank, non sym						Dist, sym						Rank, sym					
		L1		L2		L2-cxt		L1		L2		L2-cxt		L1		L2		L2-cxt	
acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain		
CSIFT	<b>91.5</b>	+2.5	79.5	+18.6	84.6	+0.9	90.9	+1.8	74.1	+13.2	84.5	+0.8	91.3	+2.3	82.4	+21.5	84.7	+1.0	
Gist	<b>84.9</b>	+1.8	83.2	+0.7	84.6	+0.1	83.8	+0.7	83.0	+0.5	84.1	-0.3	84.5	+1.5	83.2	+0.7	84.4	-0.1	
HOG	87.4	+2.9	86.8	+3.5	87.6	+0.8	87.4	+2.9	<b>88.1</b>	+4.8	86.1	-0.7	87.3	+2.9	87.8	+4.4	87.1	+0.3	
Feat.		C-PASCAL - SSL results																	
		L1		L2		L2-cxt		L1		L2		L2-cxt		L1		L2		L2-cxt	
		acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain
CSIFT	<b>24.8</b>	+0.8	18.8	+2.3	20.4	0.0	24.1	0.0	19.6	+3.1	20.7	+0.3	24.3	+0.2	20.8	+4.3	20.8	+0.3	
Gist	30.8	+2.4	30.3	+2.7	29.1	+1.0	30.4	+2.0	30.0	+2.5	29.0	+1.0	<b>32.0</b>	+3.6	31.4	+3.9	29.5	+1.4	
HOG	29.6	+10.4	24.4	+10.4	30.3	+1.5	29.5	+10.3	26.8	+12.9	30.4	+1.6	<b>32.2</b>	+13.0	29.5	+15.6	31.0	+2.1	
Har	<b>20.7</b>	+0.6	14.9	+1.8	15.4	-0.3	20.3	+0.2	16.5	+3.5	16.1	+0.4	20.5	+0.4	16.7	+3.7	16.0	+0.3	
Hess	<b>23.2</b>	+1.7	16.2	+0.9	16.6	+0.1	22.4	+0.9	17.4	+2.2	17.4	+0.9	23.1	+1.5	17.8	+2.5	17.4	+0.9	
P-BoF	<b>29.9</b>	+1.4	23.9	+3.7	23.5	+2.8	29.4	+1.0	23.8	+3.6	22.5	+1.9	<b>29.9</b>	+1.4	25.3	+5.1	23.9	+3.3	
TPBP	32.7	+3.2	29.4	+9.0	28.6	+8.2	32.0	+2.5	28.1	+7.7	26.7	+6.2	<b>33.8</b>	+4.3	30.9	+10.5	29.5	+9.0	
Feat.		AWA - SSL results																	
		L1		L2		L2-cxt		L1		L2		L2-cxt		L1		L2		L2-cxt	
		acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain	acc	gain
C-Hist	11.0	+2.6	7.6	+1.9	8.9	+0.4	10.8	+2.4	8.6	+2.9	9.1	+0.6	<b>11.2</b>	+2.8	8.7	+2.9	8.9	+0.5	
CSIFT	11.0	+3.0	8.9	+1.9	10.8	+0.5	12.8	+4.7	11.5	+4.5	12.0	+1.7	<b>13.0</b>	+5.0	11.3	+4.3	11.8	+1.5	
Gist	10.1	+2.7	10.1	+2.7	11.2	+0.3	10.8	+3.4	11.0	+3.6	<b>11.5</b>	+0.6	11.1	+3.7	11.1	+3.7	11.2	+0.4	
LSS	9.4	+2.6	6.9	+1.4	8.5	+0.3	10.9	+4.0	9.1	+3.6	9.4	+1.2	<b>11.1</b>	+4.0	9.0	+3.5	9.4	+1.2	
PHOG	7.5	+1.2	6.1	+0.7	7.2	+0.2	<b>9.4</b>	+3.2	8.0	+2.5	8.1	+1.0	<b>9.4</b>	+3.1	7.9	+2.5	8.1	+1.1	
SIFT	9.5	+1.8	8.9	+1.6	9.8	+0.6	<b>10.4</b>	+2.6	9.6	+2.3	9.9	+0.7	10.0	+2.3	9.4	+2.1	9.9	+0.7	
SURF	13.7	+4.7	8.0	+1.2	10.6	+0.3	16.3	+7.2	12.8	+6.1	13.4	+3.0	<b>16.7</b>	+7.7	12.9	+6.1	13.4	+3.0	

## 4.2 Ranking and Symmetry

Here we explicitly look at the distribution of neighbors and try to build a more intuitive and more evenly distributed structure. In particular, we would like to emphasize the symmetric relations between images when building the local neighborhood structure. First, we propose a new neighbor selection procedure to emphasize symmetric relations using the “rank as neighbor”. Second, we consider symmetry within the SSL-algorithm during graph propagation.

**Improving the structure using ranking.** In Sec. 3 for a particular distance measure and representation, we extracted *distance-based neighbors*, i.e. we look for the  $k$  images with the smallest distances to a given image, and use these images to build our local neighborhood structure.

We propose a new way of selecting neighbors, so called *rank-based neighbors*, that refines the notion of distances by emphasizing symmetric relations between images. Intuitively, we connect two images which both have the other image as

one of their nearest neighbors. More formally, we choose rank neighbors of image  $i$  as follows. We compute a first set of (distance-based) neighbors for  $i$ , and keep the one having the smallest score according to  $sc(d_i, d_j) = \tau_j(d_i) + \tau_i(d_j)$ , where  $d_i, d_j$  are the descriptors of image  $i$  and image candidate  $j$ , and  $\tau_i(d_j)$  encodes the NN rank of descriptor  $d_j$  as (distance-based) neighbor of image  $i$ . In practice, we consider only the  $l$  nearest neighbors of image  $i$  as candidates, and  $\tau_j(d_i)$  is replaced by  $\tau'_j(d_i) = \min(\tau_j(i), l)$ .  $l$  is used to narrow the search, and only needs to be large enough (here  $l=800$ ). Note that even though the function  $sc(d_j, d_i)$  is symmetric, rank-based neighborhood is not a symmetric relation.

**Improving the graph by using symmetric relations.** Sec. 3 considered *non-symmetric* (directed) graphs. Here we also look at *symmetric* (undirected) graphs. They consider incoming as well as outgoing links for propagation. There is a similar intuition behind *symmetric* graphs and rank-based NN as they both enforce more symmetric interaction between images. In the case of rank-based NN, a new structure is proposed which is potentially more effective, while for symmetric graphs, the influence of images that are too often selected as a neighbor is reduced within the existing structure.

**Experiments.** The study is divided in two parts. First, we look at the gain obtained by the structure between images using ranking, and then we study the improvement brought by both the ranking and the symmetry for the SSL results.

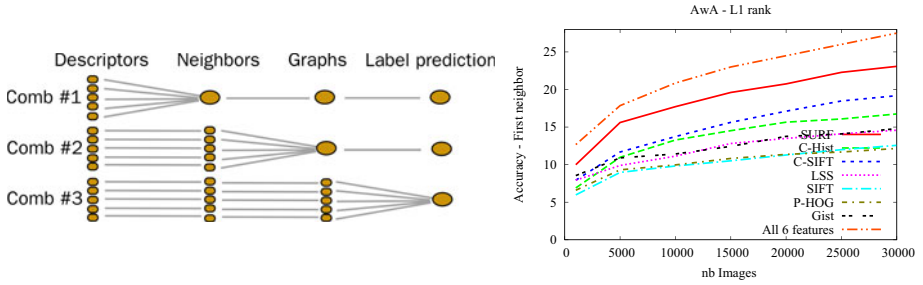
i) *Rank-based local structure.* In terms of NN quality, the rank strategy brings a consistent improvement. Over the different image descriptors, when looking at the first neighbor (1-NN quality), ETH gains 1.6% in average for L1, 9.5% for L2 and 1.9% for L2-ctxt by using the rank-based strategy. For C-PASCAL, we get 2.4% improvement for L1, 5.5% for L2 and 1.9% for L2-ctxt in average. For AWA (the results are shown in Tab. 2), the 1-NN quality for SURF is improved from 16.3% to 23% and the 10-NN quality goes from 10.5% to 14.5%. As a remark, few images were very often chosen as neighbors with the standard distance-based strategy leading to an unbalanced and unadapted structure. We observed that with the rank-based structures, this phenomena is highly reduced, which is a direct consequence of the improvement of the local structure.

Finally, L2-ctxt benefits the least from the rank NN strategy. Our intuition is that looking at the local neighborhood through the context vector allows to select more intuitive and symmetric connections.

ii) *Rank structure and symmetry for SSL results.* Tab. 3 can be directly compared with the right part of Tab. 1 and shows the following results.

First, we see for the non-symmetric case the same kind of improvement with rank-based structure as in the NN study. Tab. 1 presents results for a structure built with distance-based neighbors and a non-symmetric strategy. When comparing it with the first column of Tab. 3, we see in the gain column, that the rank brings an improvement of up to 18.6% for ETH (L2 and C-SIFT), up to 10.4% for C-PASCAL (HOG for both L1 and L2) and 4.7% for AWA (for SURF).

Next, we observe a similar, but often smaller, improvement when comparing non-symmetric graphs (right part of Tab. 1) with symmetric graphs (middle column of Tab. 3). Descriptors improving significantly with the new rank structure



**Fig. 3.** Left: SSL combination strategies. Right: 1-NN rank-based structure quality for single features and their combination on AWA

also benefit the most from the symmetric graphs. Finally, when combining the two new strategies (symmetric graphs using a rank structure, shown on the right part of Tab. 3), we obtain similar or even better results than both previous strategies. As a more general comment, rank methods (non-symmetric or symmetric) combined with L1 give nearly always the best performance and bring significant overall improvement. For C-PASCAL, TPLBP’s accuracy increases from 29.5% to **33.8%**. AWA benefits the most as the SURF descriptor improves from 9.0% accuracy to **16.7%** with a symmetric graph and a rank-based structure.

## 5 Combination

In the previous section, we have seen that we can significantly improve the local structure of the image collection for a given image representation and a given measure. In the following we will combine different features.

Feature combination has become an active area of research in the last years, and the supervised framework allows to learn the different feature contributions using the labels. Recent work [30] showed that simply averaging kernels already gives a good improvement. Consequently, both our related tasks - building the NN structure and predicting labels with SSL - should benefit from the combination of several image representations. In this section, we only look at the second task, i.e. image categorization using SSL algorithms.

**Principles.** Three graph combinations are considered (illustrated Fig. 3).

*Combination #1* assumes that the combination is done on the structure level. The different features are used to compute a single  $k$ -NN local structure. Averaging all single feature distances builds a single list of *distance* NN. A multi-feature *rank score* builds a single list of *rank* NN. This score is calculated by  $sc_{comb}(i, j) = \sum_{m \in Features} sc(d_i^m, d_j^m)$  where  $d_j^m$  is the  $m^{th}$  representation of image  $j$ , and  $sc(d_i^m, d_j^m)$  is the single feature rank score (see in Sec. 4.2).

*Combination #2* builds a graph for each feature, and forms one combined graph, the *union* graph, whose edges are the union of edges of each graph.

*Combination #3* builds as many graphs as features and propagates labels in each graph. All propagation results are combined and yield the final label.

**Table 4.** Accuracy on unlabeled data only and gain compared to the best single feature for C-PASCAL (left) and AWA (right). Results are proposed for the rank based structures, for both symmetric and non-symmetric graphs.

C-PASCAL - SSL results								AWA - SSL results							
Features	Comb	rank			rank sym			Features	Comb	rank			rank sym		
		av	var	gain	av	var	gain			av	var	gain	av	var	gain
Gist+	#1	34.0	0.8	0.0	<b>34.8</b>	0.7	+0.1	C-Hist+	#1	15.1	0.1	+1.4	<b>17.7</b>	0.2	+1.0
P-BoF+	#2	35.6	0.9	+2.9	<b>36.4</b>	0.8	+2.6	C-SIFT+	#2	<b>16.7</b>	0.1	+3.0	17.8	0.1	+1.1
TPPLBP	#3	35.8	1.0	+3.1	<b>36.7</b>	0.9	+2.9	SURF	#3	17.7	0.2	+4.0	<b>19.1</b>	0.2	+2.4
all	#1	34.9	0.8	+0.9	<b>35.7</b>	0.3	+0.9	all	#1	17.3	0.3	+3.6	<b>20.1</b>	0.3	+3.3
	#2	<b>37.2</b>	0.6	+4.5	36.8	0.5	+3.0		#2	18.1	0.2	+4.4	<b>19.2</b>	0.1	+2.5
	#3	<b>38.0</b>	0.7	+5.3	37.9	0.8	+4.1		#3	19.9	0.2	+6.2	<b>21.8</b>	0.2	+5.1

**Table 5.** Successive improvements of the local structure: best single feature with distance (top) and with rank structure and symmetric graphs (middle), and best feature combination (bottom) - for the first neighbor quality (left), and the SSL results (right).

strategy	1-NN quality			SSL-results		
	ETH	C-PASCAL	AWA	ETH	C-PASCAL	AWA
single feature	96.9	34.6	16.4	89.0	29.5	9.0
single feature + rank	97.6	38.3	23.0	91.3	33.8	16.7
multiple features + rank	98.5	45.5	27.5	94.0	38.0	21.8

Combinations #2 and #3 use multiple graphs. Each graph can either be built from distance or rank based local structures.

**Experiments.** For the C-PASCAL and AWA datasets, we combine all descriptors and the 3 best performing ones. Due to space constraints, Tab. 4 only shows the SSL results<sup>2</sup>, for L1 and for the 2 most promising strategies from the previous section, namely non-symmetric and symmetric graphs, on rank local structures. Each combination setting is considered for the 3 different combination strategies. Transductive accuracy is presented together with the gain in comparison to the best single feature within the combination.

As a first and expected conclusion, the combination of different features improves the SSL results in all settings. For C-PASCAL, the setting with all features improves the best single feature result by 5.3% reaching an accuracy of 38%. Also the AWA dataset benefits by 5.1% when combining all 7 descriptors reaching 21.8%. Second, there are only small differences between the combination methods, but combination #3 generally gives the best results.

**Summary.** If we look back at the different improvements of the local neighborhood structure we proposed in this paper, the absolute gain for each dataset is summarized in Tab. 5. In particular, SSL results are enhanced from 89% to **94%** for ETH, from 29.5% to **38%** for C-PASCAL and in the case of AWA we doubled the performance from 9% up to **21.8%** without any label. This underlines the assumption that the structure matters more than the SSL algorithm and that the structure can be improved in an unsupervised manner.

We believe that these encouraging results will be more pronounced for larger datasets. Compared to Fig. 1, Fig. 3 shows that both i) the ranking structure strategy and ii) the combination of features benefit more for larger datasets.



## 6 Conclusions

This paper explored ways of using the large amount of available image data in order to overcome inherent problems of supervised approaches. In particular, we consider methods which rely less on supervised classifiers and more on the structure of the data itself, namely the unsupervised construction of a local structure between images and the use of this structure in a SSL framework.

An important conclusion of our study is that the local structure – induced by the employed image representation, the distance measure and the number of nearest neighbors considered – matters more than the SSL algorithm. Zhu made this claim [31] together with the remark that there is only little work on the structure itself. In that sense, our study contributes to a better understanding of such structures for the tasks of object recognition and image categorization.

It is worth noting that the results obtained for the NN analysis directly translate into the corresponding performances of SSL algorithms. We indeed observed that the right set of parameters (image representation, distance measure and strategy to use it) can literally predict the SSL accuracy. On the more negative side, the overall performance obtained by the SSL algorithms is far from being satisfactory. This fits our intuition that unsupervised local structure contains some semantic information, but that the current object representations are not powerful enough for realistic datasets without supervised learning and discriminant classifiers.

To overcome these limitations we proposed different directions to improve the local structure of the dataset without any label and consequently improve the SSL results. In particular, we showed the benefits of contextual measures, symmetric relations between images, and feature combinations. Overall, a 12.8% accuracy improvement was obtained for the realistic AWA dataset without using any supervision for building the local structure.

As a conclusion, using large image collections and unsupervised local structure construction in combination with SSL algorithms is a promising direction. A generic structure can be built independently of the task, and then combined with different sets of labels. This structure can be improved by considering more suitable and complementary object and image representations, combining them, and using the information contained on the image collection topology.

## References

1. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: CVPR (2009)
2. Sivic, J., Russell, B., Zisserman, A., Freeman, W., Efros, A.: Unsupervised discovery of visual object class hierarchies. In: CVPR (2008)
3. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In: CVPR (2008)
4. Li, L.-J., Fei-Fei, L.: Optimol: Automatic online picture collection via incremental model learning. IJCV (2009)
5. Vijayanarasimhan, S., Grauman, K.: Multi-level active prediction of useful image annotations for recognition. In: NIPS (2008)

6. Russell, B., Torralba, A., Murphy, K., Freeman, W.: LabelMe: A database and web-based tool for image annotation. *IJCV* (2008)
7. Verbeek, J., Triggs, B.: Scene segmentation with CRFs learned from partially labeled images. In: *NIPS*, vol. 20 (2008)
8. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. *IJCV* (2007)
9. Leistner, C., Saffari, A., Santner, J., Bischof, H.: Semi-supervised random forests. In: *ICCV* (2009)
10. Saffari, A., Leistner, C., Bischof, H.: Regularized multi-class semi-supervised boosting. In: *CVPR* (2009)
11. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: *NIPS* (2009)
12. Chapelle, O., Schölkopf, B., Zien, A. (eds.): *Semi-Supervised Learning*. MIT Press, Cambridge (2006)
13. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: *NIPS* (2004)
14. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: *ICML* (2003)
15. Liu, W., Chang, S.: Robust multi-class transductive learning with graphs. In: *CVPR* (2009)
16. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: *CVPR* (2003)
17. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL VOC Challenge 2008 Results (2008)
18. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: *CVPR* (2009)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR* (2005)
20. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV* (2001)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *CVPR* (2006)
22. Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: *ECCV* (2008)
23. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *IJCV* (2004)
24. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* (2004)
25. van de Sande, K., Gevers, T., Snoek, C.: Evaluating color descriptors for object and scene recognition. *PAMI* (2010)
26. Hein, M., Maier, M.: Manifold denoising. In: *NIPS* (2006)
27. Bengio, Y., Delalleau, O., Le Roux, N.: Label propagation and quadratic criterion. In: [12]
28. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: *ICML* (2005)
29. Perronnin, F., Liu, Y., Renders, J.: A family of contextual measures of similarity between distributions with application to image retrieval. In: *CVPR* (2009)
30. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: *CVPR* (2009)
31. Zhu, X.: Semi-supervised learning literature survey. Technical report, UW (2005)

# Size Does Matter: Improving Object Recognition and 3D Reconstruction with Cross-Media Analysis of Image Clusters

Stephan Gammeter<sup>1</sup>, Till Quack<sup>1</sup>, David Tingdahl<sup>2</sup>, and Luc van Gool<sup>1,2</sup>

<sup>1</sup> BIWI, ETH Zürich  
{gammeter,tquack,vangool}@vision.ee.ethz.ch  
<http://www.vision.ee.ethz.ch>  
<sup>2</sup> VISICS, K.U. Leuven  
david.tingdahl@esat.kuleuven.be  
<http://www.esat.kuleuven.be/psi/visics>

**Abstract.** Most of the recent work on image-based object recognition and 3D reconstruction has focused on improving the underlying algorithms. In this paper we present a method to automatically improve the quality of the reference database, which, as we will show, also affects recognition and reconstruction performances significantly. Starting out from a reference database of clustered images we expand small clusters. This is done by exploiting cross-media information, which allows for crawling of additional images. For large clusters redundant information is removed by scene analysis. We show how these techniques make object recognition and 3D reconstruction both more efficient and more precise - we observed up to 14.8% improvement for the recognition task. Furthermore, the methods are completely data-driven and fully automatic.

**Keywords:** Image retrieval, image mining, 3D reconstruction.

## 1 Introduction

Recognition, reconstruction and analysis of 3D scenes are topics with broad coverage in the Computer Vision literature. However, in recent years the enormous amount of photos shared on the Internet has added a few new twists to these research problems. On the one hand there is the obvious challenge of scale, on the other hand there is the benefit that photos shared online usually come with meta-data in form of (geo-) tags, collateral text, user-information, *etc.* Besides the interesting research that can be done with this data, they also open doors for real-world deployments of computer vision algorithms for consumer applications, as recent examples from 3D scene browsing [1], or face recognition [2] have shown.

Consequently, a number of works have started to exploit these cross-media data in several ways [1, 3–13]. Quack *et al.* [10] have used a combination of GPS tags, textual and visual features to identify labeled objects and events in data

from community photo collections such as Flickr<sup>1</sup>. Crandall *et al.* [6] have done similar experiments, but at even larger scale (up to 10s of millions of photos) and analyzing temporal movement patterns of photographers in addition to GPS, textual and visual features. Very recently, with works such as [7], the community has started to exploit these cross-media data collections from the Web in order to build applications for auto-annotation.

Also in the 3D reconstruction field there has been a long-lasting interest in reconstruction the whole world in 3D, and not astonishingly, community photo collections nowadays serve as a data source for this purpose as well [1, 5, 12]. In spite of the different target applications, all these works have one theme in common: the underlying data structures are clusters of photos depicting the same object or scene, accompanied by some cross-modal data, such as (geo-)tags *etc.* In this work we are particularly interested in clusters of consumer photos showing “places”. Places include any geographic location, which is of interest to people, such as landmark buildings, museums, mountain peaks, *etc.* Similar to most works cited above, in a first step we also cluster images in order to identify relevant places. While attention has recently been directed towards harvesting larger and larger collections of data, in this paper we want to take a step back and look at the collected image clusters in more detail. The objective is to investigate if and how basic knowledge about the 3D scene in combination with analysis of cross-media data is helpful towards improving the quality of the database of places as well as the performance of applications building on top of the database. More precisely, we show how

- cross-media retrieval helps identifying missing information for small clusters.
- scene analysis helps removing redundant data in large clusters.
- those measures affect performance of object recognition and 3D reconstruction applications relying on the database of image clusters.

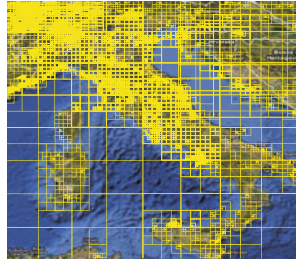
In other words, if we take the analogy of a web search engine for hypertext documents, we focus on the crawling and indexing part of the system. While in the hypertext retrieval community this topic is well documented, in the Computer Vision field most work has focussed on the retrieval side of things [14–16]. For instance, Chum *et al.* [14] could show how to improve retrieval precision using query expansion, using an algorithm which operates mainly at retrieval time.

With our improvements on the crawling and indexing stages of the pipeline, we can indirectly achieve significant improvements in an object recognition setting. We focus on the object recognition task, since there are clearly defined evaluation metrics available. In addition our contributions are valuable for unsupervised 3D reconstruction as well, however, the improvement in this application is in general less easily quantified, but easily visualized. Most importantly, for both scenarios, every proposed improvement happens offline and all the processes we show in this paper are fully automated.

The paper is structured as follows: Section 2 describes our basic methods for image cluster mining and object recognition. The core of our methods for

---

<sup>1</sup> [www.flickr.com](http://www.flickr.com)



**Fig. 1.** Geographic quadtree used for image crawling. The example shows the area around Italy. Note how the density of tiles adapts to the number of photos available, e.g. densely covering populated areas and with large tiles on the ocean.

automated cross-media cluster analysis and optimization follows in Section 3. Experiments and analysis of the effects of optimization on retrieval tasks follow in Section 4. Section 5 concludes the paper.

## 2 Mining and Recognition of Objects

As discussed in the introduction, harvesting photos from online services for landmark mining, recognition or 3D reconstruction has been addressed in a number of recent works. We build on some of those ideas in order to construct our own image mining pipeline. We also introduce the object recognition methods, which we apply on top of the mined data.

### 2.1 Object Mining

Several ways have been proposed to collect data from online photo collections in order to solve computer vision tasks. They either start out by querying with certain keywords such as "Rome", "Venice" [1, 12, 17, 18], or with collecting geo-tagged photos [6, 10]. For bootstrapping our system we chose the latter strategy.

In order to harvest photos from Flickr based on their geo-tags, we overlay several geographic quad-trees over the world and retrieve the number of photos in each tile. Each of the trees is initialized by a country's geographic bounding box coordinates. Recursively this initial area is then subdivided as follows. We retrieve the number of photos in the current area from the Flickr API. When the number of photos is higher than a threshold (250 in our implementation), we split the area into 4 tiles of equal size and repeat the process for each tile. The recursion stops when the threshold for the number of photos is reached. In addition, the dimension of the tile in meters also serves as a second stopping criterion: the process returns when the tile's extent is less than 200m (on the smaller side). The outcome of this is shown in Fig. 1. Photos are then downloaded for all child leaves, and the photo clustering is also distributed based on the

child leaves of the geographic quadtree. For clustering photos, we then proceed as proposed in [10] in three steps

1. Match photos pair-wise using local image features (we use SURF [19]).
2. Build a set of image similarity matrices. We create one matrix per geographic leaf tile. The similarity is the number of inlying matches after RANSAC filtering of feature matches for each similar image pair.
3. Cluster the photos using single-link hierarchical agglomerative clustering.

For each cluster we keep its photos including their meta data (tags, titles, user information *etc.*) for further processing. Very similar to [10] we observed that the image clusters usually represent one common object, but covered with photos from various viewpoints and under various lighting conditions *etc.* Thus, we think of each cluster representing one particular object and consider the images of a cluster to form an exemplar based object model.

Qualitatively, we think our crawling method ends-up with very similar data like [10], but is significantly more efficient ([10] scans the world in evenly distributed tiles of equal size, in effect querying a lot of empty cells unnecessarily.) We believe our crawling approach is also beneficial over [6], since we can split the clustering problem into smaller parts, and the tree based approach is directly “pulled” towards densely populated areas already while collecting the data. In contrast, [6] is one huge clustering problem. Finally we crawled a significantly larger dataset than [7] with our quadtree method (17 million images w.r.t. 4 million), to be able to compare our results in terms of object recognition with theirs as a baseline, for the remainder of this paper we conduct all our analysis on the same data (the dataset is available from the authors web-site).

## 2.2 Object Recognition

Given a query image depicting a landmark, the goal is now to identify and label this object based on the information aggregated in our reference database of image clusters. This task is very similar to the one recently posed by Gammerter *et al.* [7]. (In contrast to image/object retrieval [20–22], where the expected outcome is a ranked list of similar images or images showing the same object as the query, sorted by similarity).

Much like the work of [7], at the lowest level, our object recognition system builds on “standard” visual word based image retrieval. Local image features [19] are clustered into a visual vocabulary of 1 million visual words using approximate  $k$ -Means (AKM) [21]. An initial *top-n list* of the  $n$  most similar images in the database in terms of set intersection is efficiently computed using an inverted file structure. We then use RANSAC to estimate a homography between the query image and every image in the *top-n list*. Candidate images for which the RANSAC estimate yields less inliers than a threshold (13 in our implementation) are discarded. We then simply let the image with the highest number of inliers to identify the object in the query image. This is in contrast to [7], where the images in the filtered *top-n list* are used to vote for “their” object.

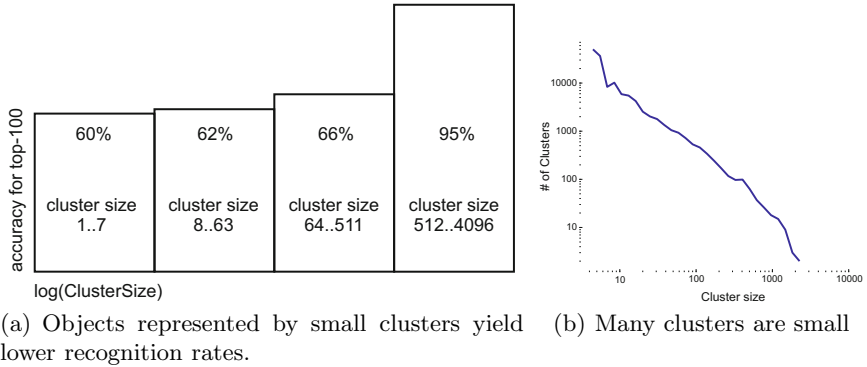


Fig. 2.

### 3 Cross-Media Cluster Analysis

The main object of study for the remainder of this paper are the image clusters mined from the Internet as described in the preceding section. Given our target applications — object recognition or 3D reconstruction — we can now analyze and improve the image clusters in several aspects. The first and most obvious aspect is cluster size. Intuitively, objects which are represented by a smaller cluster should be more difficult to recognize, since they may lack images taken from an important viewpoint. Fig. 2(a) shows a (histogram) plot of the cluster size versus recognition rate. It confirms that recognition tends to be more successful for larger image clusters. (Detailed results for recognition are given in Section 4 of this paper.) Further, as illustrated in Figure 2(b), it seems that the cluster size distribution follows a power law:  $p(\text{ClusterSize}) \propto \frac{1}{\text{ClusterSize}^\alpha}$  with a maximum likelihood estimate of  $\alpha_{MLE} = 1.41$ . Such distributions are extremely heavy tailed, and thus imply several characteristics. For instance, one should note that it is unreasonable to consider an average cluster size, since the expectation value diverges for  $\alpha \leq 2$ . Further, from the power law distribution also follows that the majority of clusters is small, but due to the heavy tail quite a few clusters are disproportionally large. It stands to reason that these extremely large clusters carry a large amount of redundant information. Thus, in the following, we investigate the effect of expanding small clusters with additional (non geo-tagged) images, and propose strategies for reducing redundant information contained in very large image clusters.

#### 3.1 Expansion of Small Clusters

Even though an increasing number of digital images shared online contain geo-tags, owning a GPS-equipped camera is still not standard today. Consequently, a significant fraction of clusters mined using an approach relying on geo-tags, consists only of a handful of images (Fig. 2(b)). In fact, in our dataset 81% of all clusters contain 10 images or less. For some places this is simply because they are not



**Fig. 3.** Cross-media expansion of image clusters: 1) starting out from clusters of images (clustered by their visual similarity with the help of geo-tags for efficiency), we use itemset mining to generate text queries from frequent tags. 2) in order to retrieve additional images thus expanding the image cluster with additional information, 3) and finish with a verifying matching based on visual similarity. We also show the Cluster Match Rate (CMR) for each itemset query (see Section 3.2)

popular enough. Note that with keyword based mining we would not have been able to find such rare objects in the first place — a list of terms that extensive that it covers such locations is simply not available. But even for much-visited locations many images can lack GPS tags, if the location is *e.g.* inside a building. In order to enrich such small clusters, we propose to use a cross-media crawling method. First, text queries are generated using the tags associated with an existing image cluster. To that end, we follow the approach taken by [10], where text queries are automatically created from the meta-data of the photos in each cluster. They then use these queries for crawling Wikipedia articles intended to serve as descriptions for image clusters. In order to generate the text queries automatically, the authors propose to use itemset mining [23] to form frequent combinations of tags for each cluster. We follow the same approach, but query the WWW for images instead for Wikipedia articles. For the remainder of this paper we call these automatically generated text queries *itemset queries*. The itemset queries are used to query common *Google* for additional photos. The retrieved images are then matched against the images inside the cluster, again by estimating a Homography using RANSAC and SURF [19] features. Matching images are added to the cluster. Match vs. no match is determined based on an inlier threshold of 15 feature correspondences. This procedure is illustrated in Fig. 3.

### 3.2 Efficient Itemset Query Selection

It turns out, that for a surprisingly large amount of clusters additional images can be retrieved (96% of clusters in our test dataset have been expanded by at least one image). Furthermore, one should note that as shown in Fig. 2(a) this procedure is more likely to be successful for larger clusters than for smaller ones. Obviously, when applying such an automatic query generation approach for a large amount of data with many clusters, the number of text queries can

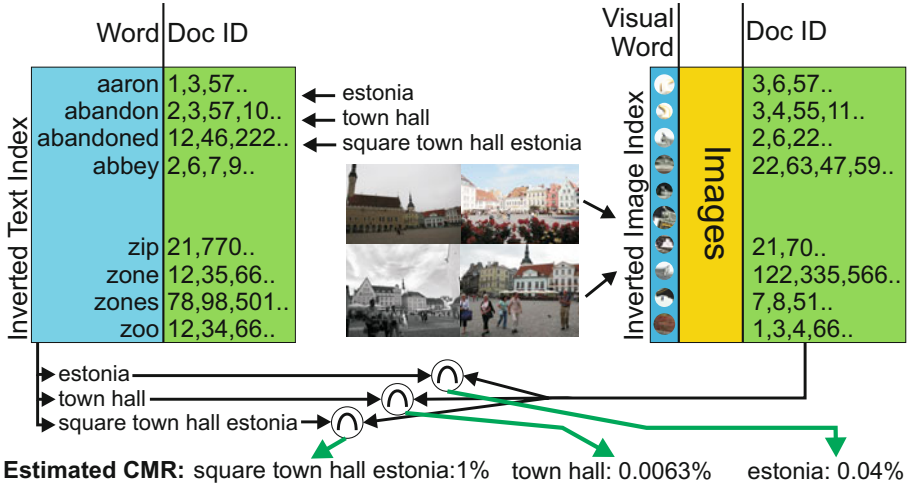


reach a level, where efficiency considerations become crucial. (Each cluster can generate dozens or even up to hundreds of different itemset queries). Unlike other resources like bandwidth or storage, the amount of HTTP requests that can be made to a public service like Google is often limited. Furthermore, results from search engines are returned aggregated to pages. Each page usually contains only about 20 images and requires one additional HTTP request to retrieve it. While the prices of resources like computation power (Moore’s Law), bandwidth (Nielsen’s Law) and storage (Kryder’s Law) drop exponentially over time, this most likely does not imply the same exponential increase in the number of queries that can be made to search engines. (They are already confronted with a rapidly growing user base.) So, unless one has the resources to crawl the entire Internet in order to avoid public search engines, it is of great interest to minimize the number of queries required. However, if an itemset query is not very specific (e.g. “town hall”, compare Fig. 3), it might lead to the retrieval of a large number of images, which do not have anything in common with the object in the cluster, and consequently won’t match to its images. In other words, to be efficient, we have to find a way to automatically select itemset queries which have a higher probability of returning relevant images.

As a basic measure for how successful an itemset query is in retrieving additional images of the object, we define first the *cluster matching rate (CMR)*.

$$CMR = \frac{\# \text{ Matching images}}{\# \text{ Retrieved images}} \quad (1)$$

This is a straightforward choice, which records for a given itemset query the fraction of retrieved images that match to the images in the database cluster. While CMR is useful to determine the quality of an itemset query once all images have already been retrieved and matched, an efficient approach should discard itemset queries with low CMR well before that. This could entail estimating the CMR, which in turn would require in the order of  $(1/CMR - 1)$  images. Thus, the lower the CMR of an itemset query, the more images we would have to download before we can reject it. By comparing the improvement in recognition quality on the test set when considering all queries vs. the improvement when only accepting queries with a CMR above a given threshold, we find that the largest improvement comes from queries with a CMR between 0.01 and 0.1. This is shown in Fig. 7. In other words, we might have to download at least 100 images before we can safely reject any itemset query. We can, however, exploit an observation made by [10, 24]. The authors used text queries in order to retrieve Wikipedia articles intended as descriptions for the image clusters. The trick they came up with, is to verify the retrieval result by matching images from the articles to the source cluster. They found that itemset queries yielding articles containing images matching the cluster have a higher probability of yielding matching images from other sources as well. This could be a crude indicator to a-priori assess an itemset query’s CMR. In order to test this hypothesis, we downloaded and indexed a dump of all English Wikipedia articles and their images. Then, as illustrated in Fig. 4 for any given cluster, we query both the text index with the itemset queries and the image index with the cluster’s

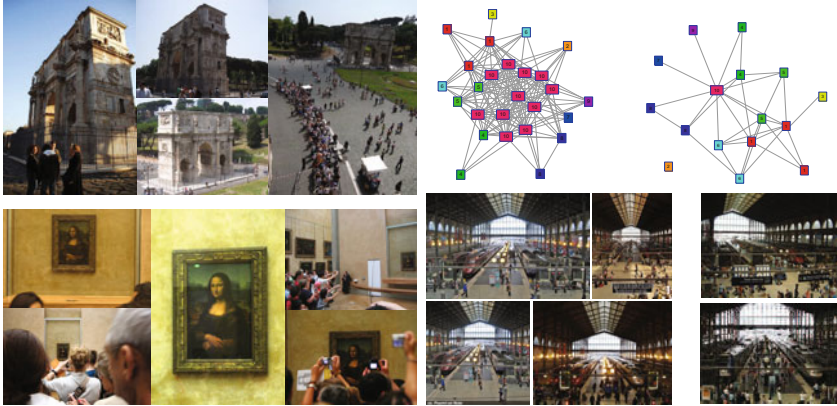


**Fig. 4.** A priori estimation of CMR using a local copy of Wikipedia. An inverted text index and an image index are queried simultaneously. The result sets are intersected in order to determine if the text could have yielded any useful images

images. The text index returns ids of Wikipedia articles with words matching the itemset query, while the image index returns ids of Wikipedia articles with images matching the cluster images. The two result sets are then intersected. The ratio of the number in the intersected set and the number of elements in the set returned by the text index can be taken as a crude estimate of the CMR. This estimation is shown as EST value in Fig. 3. With this measure at hand, we are able to discard a significant amount of irrelevant itemset queries early on.

### 3.3 Reduction of Large Clusters

While small objects that are only modeled by few images in their respective image clusters are more difficult to recognize, having too much data is not a blessing either. Unusually large amounts of photos are often collected at popular tourist destinations such as Notre Dame de Paris, or the Eiffel Tower. Many of these photos contain redundant information, which in an image retrieval scenario, unnecessarily increase the size of the inverted index. Furthermore, since our method from Section 3.1 allows for augmenting almost any cluster by an arbitrary amount of images, we desire to find a method that purges the redundant information, while leaving complementary information untouched. Note that it is a-priori also unclear what “a good” number of images would be for an arbitrary cluster, since it strongly depends on the 3D scene structure of the given object. This is illustrated in Fig. 5. The object on the top left is a free standing structure which can be photographed from arbitrary viewpoints, so an image cluster which serves as a model for this object has to contain many images. In contrast, the example on the bottom left is the extreme case of a painting in a museum, which can be seen from a small number of viewpoints only, so fewer reference



**Fig. 5.** Left column: example of a free-standing 3D object which can be photographed from many viewpoints (top) vs. one which is visible nearly from only a single viewpoint (and even only 2-dimensional in this particular case). Right column: example of cluster reduction. the full single-linked matching graph is shown on the top left. A complete-linked section which is removed on the top right. Bottom left: images from the removed complete-link segment. Bottom right: images which stay in the cluster.

images are necessary to “describe” the object. In fact, while 3D scene structure makes it impossible to generalize to a “good” cluster size, it is at the same time key to attack the problem of extraordinary cluster size. It turns out, that with some simple 3D scene analysis we can compact the clusters in both an effective and scalable manner. Remember, that the image clusters were created using single-link clustering (Section 2). We now decompose these single-link clusters into several overlapping complete-link clusters. Note the definition of single-link and complete-link criteria in hierarchical agglomerative clustering [25]

$$\text{single-link: } d_{AB} = \min_{i \in A, j \in B} d_{ij} \quad \text{complete-link: } d_{AB} = \max_{i \in A, j \in B} d_{ij}$$

where for clusters  $A$  and  $B$  the indices  $i, j$  run over the images in the clusters and  $d_{ij}$  is the image distance measure that is proportional to the inverse of the number of inliers. Complete-link requires that all image pairs in a segment are fully connected to each other. In our setting this is the case if all image pairs match to each other, which means that they are all taken from a very similar viewpoint. This procedure is illustrated in Fig. 5. Then, for every complete-link cluster with more than 3 nodes, we find the node with the minimum edge-weight-sum (*i.e.* the image most similar to all its neighbors) and remove all other nodes. In essence it is an idea similar to the scene graph in [26], but can here be derived with standard tools using the already calculated distance matrix.

When we remove these highly similar images from the index we automatically remove highly redundant information, while guaranteeing that we keep relevant data. As demonstrated in the experiments in section 4.1 this procedure reduces the index size for retrieval tasks significantly, without affecting precision.

## 4 Experiments and Results

For all our experiments we used the dataset of [7], which can be obtained from the authors website. The dataset consists of roughly 1 Million images from Flickr that were clustered into 63'232 objects and a test set of 676 images which are associated with 170 of the 63'232 objects. The goal is to correctly identify which object in the database is shown in the images of the test set. The percentage of images correctly associated with their object serves as an evaluation metric. We first report evaluation results on overall recognition performance including the overall effects of cluster expansion and cluster reduction. Finally, we demonstrate that our additionally mined images can be vital in 3D reconstruction.

### 4.1 Object Recognition

We compare our object recognition system to the one of [7]. On their benchmark dataset we achieve similar baseline performance, as shown in Fig. 6. Adhering to the original evaluation protocol of [7] we consider the percentage of test images for which the correct object is returned in its top- $n$  candidate list vs. the toplist size  $n$ . This is an upper limit for the recognition rate after geometric verification. We then applied our cluster expansion and reduction methods to the image clusters in the benchmark dataset. For each of the 170 clusters in the testset we generated itemset queries in order to retrieve additional images for cluster expansion according to the methods described in Section 3.1. We carried out experiments with 3 major image search engines and found that using *Google* yielded the best results. For every itemset query we retrieve the first 420 images returned by *Google* to expand our object models. Fig. 6 clearly shows that expanding clusters substantially improves recognition. However, since we only expand clusters that are relevant to the test dataset, we created an unfair situation: the expanded clusters now have a higher probability of randomly occurring in a top- $n$  list. We thus plot the chance level in Fig. 6 (dashed lines) for each expanded index. The comparison highlights that the observed improvement is not simply an artifact of an increased chance level.

A summary of the achieved improvements over the baseline is given in Table 1. The first two columns show cluster retrieval results with bag of visual words lookup for finding the correct cluster in the top  $n$  ranked results. The third column shows results for identifying the correct object/cluster on the first rank, using geometric verification. To that end the top ranked 1000 results after bag of words lookup were verified by estimating a Homography mapping between query and retrieved images using RANSAC. For this last task, we achieve 14.8% improvement over [7], when using our cluster expansion method. We also applied the reduction strategy from Section 3.3 to the baseline index, as well as to the expanded index. In both cases we find that our strategy for “purging” unnecessary images does not significantly influence recognition quality as demonstrated in Fig. 6. However, it reduces the inverted index file size significantly, as shown in Table 2. One can also observe that the relative reduction in size is much larger for the expanded index. This is due to the fact, that retrieving additional images via itemset queries more often leads to duplicates or

**Table 1.** Absolute number of testsets with a correct cluster within the top-100 and top-1000 list. The last column is the absolute number if test images correctly labeled after geometric verification of the top-1000 list.

Description	top-100	top-1000	top-1 Geo.Ver.
Baseline	63.4%	78.6%	73.52%
Expanded	73.0%	86.1%	78.1%

**Table 2.** Index size comparison for indices built from the original clusters vs. reduced clusters

Description	Original	Reduced
Baseline	1.5GB	1.3GB (-13%)
Expanded	2.1GB	1.5GB (-29%)

near duplicate images compared to images retrieved using GPS queries during the initial crawling of clusters.

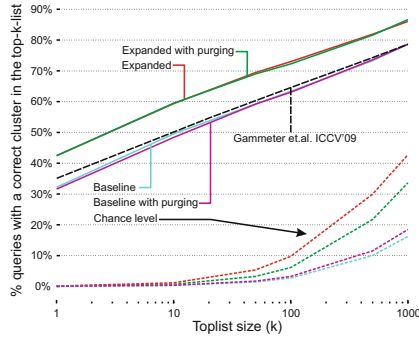
## 4.2 Efficient Itemset Query Selection

In total 2030 itemset queries were generated for the testset of 170 image clusters. As mentioned in Section 3.2, on the fly estimation of CMR based on retrieved images is not beneficial, since at least 100 images have to be retrieved before an itemset query can be safely discarded. However we can use the *estimated CMR* (c.f. Section 3.2) as an indicator if an itemset query is useful or not. This is demonstrated in Fig. 7. We found that if we do not discard itemset queries with an *estimated CMR* above 0.01% we retain about 75% of the original improvement. For the test dataset only 40% of all queries fulfill this requirement, however as visible in Fig. 7 these queries alone are responsible for the 75% improvement in recognition quality.

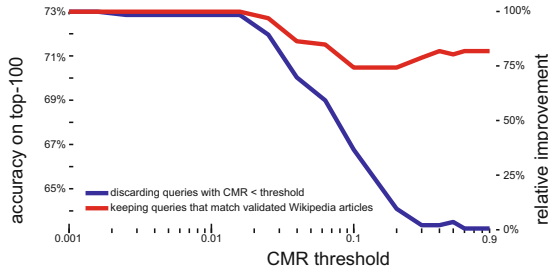
## 4.3 3D Reconstruction

So far we focussed on demonstrating the outcome of the proposed cluster expansion and reduction methods on an object recognition task. As mentioned earlier, this is due to the easy quantification of the evaluation. However, the same methods can also be beneficial in a 3D reconstruction scenario. The outcome of image based 3D reconstruction is highly dependent on the images used as input. In essence, a large number of high-resolution images taken from a wide variety of viewpoints is desired.

That a simple keyword search or geographic query yields enough images for a decent reconstruction of an arbitrary object is far from a given. Such a strategy in fact only works for a fraction of all landmarks. Even for popular sites, manual keyword search is not trivial, because it is not feasible to efficiently come up with so many appropriate keywords. For less famous landmarks, the situation is



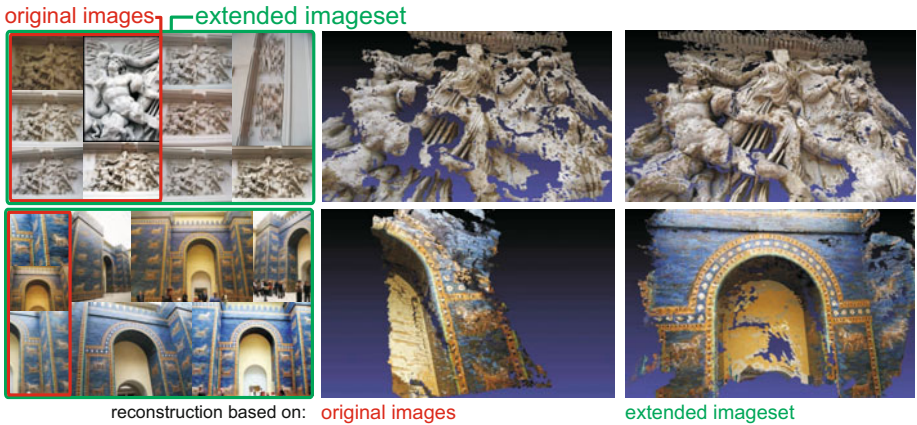
**Fig. 6.** Expanding clusters with additional images from Google significantly improves top- $n$  score. Reduction of clusters (with “purging” of complete-link segments) does not affect performance, neither for the baseline nor for previously expanded clusters. Improvements can not be attributed to chance, as the dashed lines show.



**Fig. 7.** Considering top-100 accuracy, we compare the overall improvement to the baseline obtained when considering only queries with a CMR above a certain threshold (blue line). The red line shows what happens if we discard itemset queries if and only if their CMR is below a certain threshold *and* their *estimated* CMR is below 0.01%.

even more dire. Even a keyword search with the precise description of the object may not yield enough useful images nor would GPS-based retrieval.

In such cases every single image matters, and a couple of additional images of high quality may dramatically change the outcome of the reconstruction. In this section, we briefly demonstrate with two examples that our cluster expansion method yields additional images crucial for 3D reconstruction. Using the publicly available ARC3D [27] reconstruction tool, we compare the 3D reconstruction of the originally mined image clusters of [7] to the reconstruction based on our expanded clusters. From a set of uncalibrated images, ARC3D generates dense, textured depth maps for each image. Input images are uploaded and processing is performed remotely on a cluster, so that results can be obtained within short time. As demonstrated in Fig. 8, additionally mined images clearly help in reconstructing more complete 3D models.



**Fig. 8.** Unsupervised 3D reconstruction using ARC3D. The first row shows the initial clusters (red box) and the additional mined images (green box). The second row shows the 3D reconstruction only using the initial image set. Reconstruction based on the extended set is shown in the third row. As can be seen, our additionally mined images clearly make the reconstructed 3D models more complete

## 5 Conclusion

We have shown a fully automated cross-media method to improve the quality of reference databases for object recognition. Small image clusters were enriched with additional information by automatically generating text-queries from image meta-data. Redundant information was purged from large clusters by a simple graph based approach. The combination results in better performance and higher efficiency (in index size) for object recognition tasks on recent benchmark data for object instance recognition. We have also shown that it is possible to exploit the wisdom of crowds to a-priori determine if a potential text query may be useful for retrieving additional images. Finally, while this paper focussed on object recognition, the cluster expansion method would be also valuable for unsupervised 3D reconstruction.

**Acknowledgments.** We would like to thank the *Swiss National Science Foundation Project IM2* and *Google* for their support of this research.

## References

1. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: Exploring photo collections in 3rd. *ACM Trans. on Graphics* 25 (2006)
2. Stone, Z., Zickler, T., Darrell, T.: Autotagging facebook: Social network context improves photo annotation. In: *IEEE Workshop on Internet Vision CVPR 2008* (2008)
3. Zheng, Y., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T.S., Neven, H.: Tour the world: Building a web-scale landmark recognition engine. In: *CVPR* (2009)
4. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections (2009)

5. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: ICCV (2009)
6. Crandall, D., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: WWW '09: Proceedings of the 18th International Conference on World Wide Web (2009)
7. Gammeter, S., Bossard, L., Quack, T., Van Gool, L.: I know what you did last summer: object level auto-annotation of holiday snaps. In: ICCV (2009)
8. Hays, J., Efros, A.A.: Im2gps: estimating geographic information from a single image. In: CVPR (2008)
9. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A.A., Hertzmann, A.: Image sequence geolocation with human travel priors. In: ICCV (2009)
10. Quack, T., Leibe, B., Van Gool, L.: World-scale mining of objects and events from community photo collections. In: CIVR (2008)
11. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. In: ICCV (2007)
12. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: ICCV (2007)
13. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. PAMI 30, 1958–1970 (2008)
14. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: CVPR (2007)
15. Chum, O., Matas, J.: Web scale image clustering. Technical report, Czech Technical University Prague (2008)
16. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
17. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 427–440. Springer, Heidelberg (2008)
18. Philbin, J., Zisserman, A.: Object mining using a matching graph on very large image collections. In: ICVGIP (2008)
19. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
20. Nistér, D., Stewénus, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
22. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: ICCV (2003)
23. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: SIGMOD (1993)
24. Gool, L.V., Breitenstein, M.D., Gammeter, S., Grabner, H., Quack, T.: Mining from large image sets. In: CIVR (2009)
25. Webb, A.: Statistical Pattern Recognition, 2nd edn. Wiley, Chichester (2002)
26. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR (2008)
27. Vergauwen, M., Van Gool, L.: Web-based 3rd reconstruction service. MVA 17, 411–426 (2006)



# Avoiding Confusing Features in Place Recognition

Jan Knopp<sup>1</sup>, Josef Sivic<sup>2</sup>, and Tomas Pajdla<sup>3</sup>

<sup>1</sup> VISICS, ESAT-PSI, K.U. Leuven, Belgium

<sup>2</sup> INRIA, WILLOW project, Ecole Normale Supérieure, Paris, France\*

<sup>3</sup> Center for Machine Perception, Czech Technical University in Prague

**Abstract.** We seek to recognize the place depicted in a query image using a database of “street side” images annotated with geolocation information. This is a challenging task due to changes in scale, viewpoint and lighting between the query and the images in the database. One of the key problems in place recognition is the presence of objects such as trees or road markings, which frequently occur in the database and hence cause significant confusion between different places. As the main contribution, we show how to avoid features leading to *confusion* of particular places by using geotags attached to database images as a form of supervision. We develop a method for automatic detection of image-specific and spatially-localized groups of confusing features, and demonstrate that suppressing them significantly improves place recognition performance while reducing the database size. We show the method combines well with the state of the art bag-of-features model including query expansion, and demonstrate place recognition that generalizes over wide range of viewpoints and lighting conditions. Results are shown on a geotagged database of over 17K images of Paris downloaded from Google Street View.

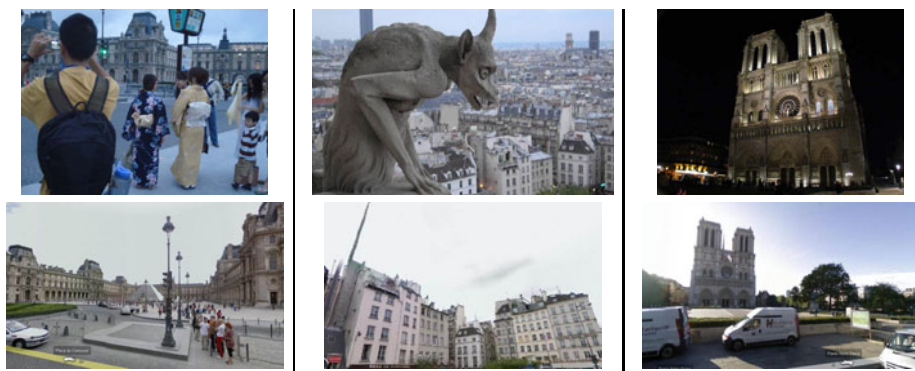
## 1 Introduction

Map-based collections of street side imagery, such as Google StreetView [1] or Microsoft StreetSide [2] open-up the possibility of image-based place recognition. Given the query image of a particular street or a building facade, the objective is to find one or more images in the geotagged database *depicting the same place*. We define “place” as the 3D structure visible in the query image, rather than the actual camera location of the query [3]. Images showing (a part of) the same 3D structure may, and often have, very different camera locations, as illustrated in the middle column of figure 1.

The ability to visually recognize the place depicted in an image has a range of exciting applications such as: (i) automatic registration of consumer photographs with maps [4], (ii) transferring place-specific annotations, such as landmark information, to the query image [5,6], or (iii) finding common structures between

---

\* Laboratoire d’Informatique de l’Ecole Normale Supérieure (CNRS/ENS/INRIA UMR 8548).



**Fig. 1.** Examples of visual place recognition results. Given a query image (top) of an unknown place, the goal is to find an image from a geotagged database of street side imagery (bottom), depicting the same place as the query.

images for large scale 3D reconstruction [7]. In addition, it is an important first step towards estimating the actual query image camera location using structure from motion techniques [8,9,7].

Place recognition is an extremely challenging task as the query image and images available in the database might show the same place imaged at a different scale, from a different viewpoint or under different illumination conditions. An additional key challenge is the self-similarity of images of different places: the image database may contain objects, such as trees, road markings or window blinds, which occur at many places and hence are not representative for any particular place. In turn, such objects significantly confuse the recognition process.

As the main contribution of this work, we develop a method for automatically detecting such “confusing objects” and demonstrate that removing them from the database can significantly improve the place recognition performance. To achieve this, we employ the efficient bag-of-visual-words [10,11] approach with large vocabularies and fast spatial matching, previously used for object retrieval in large unstructured image collections [12,13]. However, in contrast to generic object retrieval, the place recognition database is structured: images depict a consistent 3D world and are labelled with geolocation information. We take advantage of this additional information and use the available geotags as a form of supervision providing us with large amounts of negative training data since images from far away locations cannot depict the same place. In particular, we detect, in each database image, spatially localized groups of local invariant features, which are matched to images far from the geospatial location of the database image. The result is a segmentation of each image into a “confusing layer”, represented by groups of spatially localized invariant features occurring at other places in the database, and a layer discriminating the particular place from other places in the database. Further, we demonstrate that suppressing such confusing features significantly improves place recognition performance while reducing the database size.

To achieve successful visual place recognition the image database has to be representative: (i) all places need to be covered and (ii) each place should be captured under wide range of imaging conditions. For this purpose we combine two types of visual data: (i) street-side imagery from Google street-view which has good coverage and provides accurate geo-locations; and (ii) user-generated imagery from a photo-sharing website Panoramio, which depicts places under varying imaging conditions (such as different times of the day or different seasons), but is biased towards popular places and its geotags are typically noisy. We show place recognition results on a challenging database of 17K images of central Paris automatically downloaded from Google Street View expanded with 8K images from the photo-sharing website Panoramio.

## 1.1 Related Work

Most previous work on image-based place recognition focused on small scale settings [14,15,16]. More recently, Cummins and Newman [17] described an appearance-only simultaneous localization and mapping (SLAM) system, based on the bag-of-features representation, capturing correlations between different visual words. They show place recognition results on a dataset of more than 100,000 omni-directional images captured along a 1,000 km route, but do not attempt to detect or remove confusing features. Schindler *et al.* [3] proposed an information theoretic criterion for choosing informative features for each location, and build vocabulary trees [18] for location recognition in a database of 30,000 images. However, their approach relies on significant visual overlap between spatially close-by database images, effectively providing positive “training data” for each location. In contrast, our method measures only statistics of mismatched features and requires only negative training data in the form of highly ranked mismatched images for a particular location.

Large databases of several millions of geotagged Flickr images were recently used for coarse-level image localization. Hays and Efros [19] achieve coarse-level localization on the level of continents and cities using category-level scene matching. Li *et al.* [6] discover distinct but coarse-level landmarks (such as an entire city square) as places with high concentration of geotagged Flickr images and build image-level classifiers to distinguish landmarks from each other. In contrast, we address the complementary task of matching particular places in street-side imagery, use multi-view spatial constraints and require establishing visual correspondence between the query and the database image.

Community photo-collections (such as Flickr) are now often used in computer vision tasks with the focus on clustering [20,21,5], 3D modelling [9,7] and summarization [22]. In contrast, we combine images from a community photo-collection with street-side imagery to improve place recognition performance.

The task of place recognition is similar to object retrieval from large unstructured image collections [20,23,18,13,24], and we build on this work. However, we propose to detect and suppress confusing features taking a strong advantage of the structured nature of the geolocalized street side imagery.

Finally, the task of confuser detection has some similarities with the task of feature selection in category-level recognition [25,26,27] and retrieval [28,29,30]. These methods typically learn discriminative features from clean labelled data in the Caltech-101 like setup. We address the detection and suppression of spatially localized groups of confusing (rather than discriminative) features in the absence of positive (matched) training examples, which are not directly available in the geo-referenced image collection. In addition, we focus on matching particular places under viewpoint and lighting variations, and in a significant amount of background clutter.

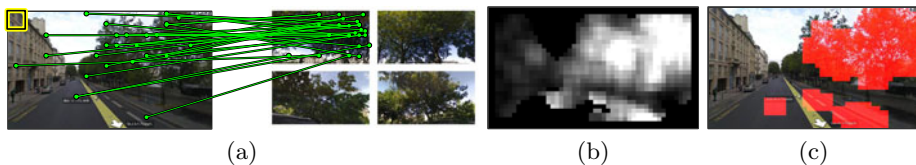
The remainder of the paper is organized as follows. Section 2 reviews the baseline place recognition algorithm based on state-of-the-art bag-of-features object retrieval techniques. In section 3 we describe the proposed method for detection of spatially localized groups of confusing features and in section 4 we outline how the detected confusers are avoided in large scale place matching. Finally, section 5 describes the collected place recognition datasets and experimentally evaluates the benefits of suppressing confusers.

## 2 Baseline Place Recognition with Geometric Verification

We have implemented a two-stage place recognition approach based on state-of-the-art techniques used in large scale image and object retrieval [18,13]. In the first stage, the goal is to efficiently find a small set of candidate images (50) from the entire geotagged database, which are likely to depict the correct place. This is achieved by employing the bag-of-visual-words image representation and fast matching techniques based on inverted file indexing. In the second verification stage, the candidate images are re-ranked taking into account the spatial layout of local quantized image features. In the following we describe our image representation and give details of the implementation of the two image matching stages.

*Image representation:* We extract SURF [31] features from each image. They are fast to extract (under one second per image), and we have found them to perform well for place recognition in comparison with affine invariant features frequently used for large-scale image retrieval [23,18,13] (experiments not shown in the paper). The extracted features are then quantized into a vocabulary of 100K visual words. The vocabulary is built from a subset of 2942 images (about 6M features) of the geotagged image database using the approximate k-means algorithm [32,13]. Note that as opposed to image retrieval, where generic vocabularies trained from a separate training dataset have been recently used [23], in the context of location recognition a vocabulary can be trained for a particular set of locations, such as a district in a city.

*Initial retrieval of candidate places:* Similar to [13], both the query and database images are represented using tf-idf [33] weighted visual word vectors and the similarity between the query and each database vector is measured using the



**Fig. 2.** Detection of place-specific confusing regions. (a) Features in each database image are matched with features of similar images at geospatially far away locations (illustration of matches to only one image is shown). (b) Confusion score is computed in a sliding window manner, locally counting the proportion of mismatched features. Brightness indicates high confusion. (c) An image is segmented into a “confusing layer” (indicated by red overlay), and a layer (the rest of the image) discriminating the particular place from other places in the database.

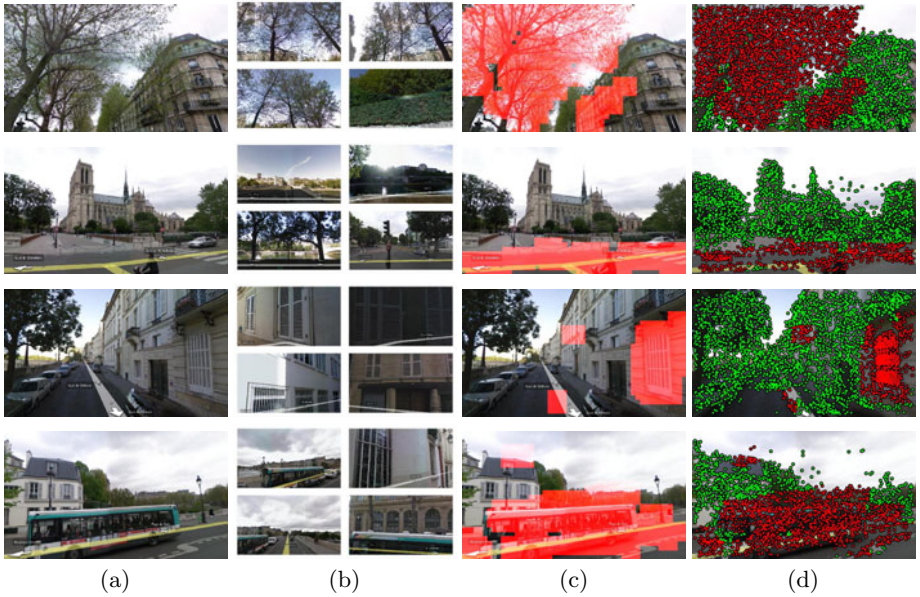
normalized scalar product. The tf-idf weights are estimated from the entire geotagged database. This type of image matching has been shown to perform near real-time matching in datasets of 1M images [23,18,13]. After this initial retrieval stage we retain the top 50 images ranked by the similarity score.

*Filtering by spatial verification:* In the second stage we filter the candidate set using a test on consistency of spatial layout of local image features. We assume that the 3D structure visible in the query image and each candidate image can be approximated by a small number of planes (1-5) and fit multiple homographies using RANSAC with local optimization [34]. The piecewise planar approximation has the benefit of increased efficiency and has been shown to perform well for matching in urban environments [13]. The candidate images are then re-ranked based on the number of inliers.

*Enhancing street-side imagery with additional photographs:* In image retrieval query expansion has been shown to significantly improve retrieval performance by enhancing the original query using visual words from spatially-verified images in the database [12]. Here, we perform query expansion using a collection of images downloaded from a photo-sharing site and details of this data will be given in section 5. These images are not necessarily geotagged, but might contain multiple images of the same places captured by different photographers from different viewpoints or different lighting conditions. The place recognition algorithm then proceeds in two steps. First the query image is expanded by matching to the non-geotagged database. Second, the enhanced query image is used for the place recognition query to the geotagged database. We implement the “average query expansion” described in [12].

### 3 Detecting Spatially Localized Groups of Confusing Features

Locations in city-street image databases contain significant amount of features on objects like trees or road markings, which are not informative for recognizing



**Fig. 3.** Examples of detected confusing regions which are obtained by finding local features in original image (a) frequently mismatched to similar images of different places shown in (b). (c) Detected confusing image regions. (d) Features within the confusing regions are erased (red) and the rest of features are kept (green). Note that confusing regions are spatially localized and fairly well correspond to real-world objects, such as trees, road, bus or a window blind. Note also the different geospatial scale of the detected “confusing objects”: trees or pavement (top two rows) might appear anywhere in the world; a particular type of window blinds (3rd row) might be common only in France; and the shown type of bus (bottom row) might appear only in Paris streets. Confusing features are also place specific: trees deemed confusing at one place, might not be detected as confusing at another place, depending on the content of the rest of the image. Note also that confusion score depends on the number of detected features. Regions with no features, such as sky, are not detected.

a particular place since they appear frequently throughout the city. This is an important problem as such features pollute the visual word vectors and can cause significant confusion between different places. To address this issue we focus in this section on automatically detecting such regions. To achieve this, we use the fact that *an image of a particular place should not match well to other images at far away locations*. The details of the approach are given next.

*Local confusion score:* For each database image  $I$ , we first find a set  $\{I_n\}$  of top  $n$  “confusing” images from the geotagged database. This is achieved by retrieving top matching images using fast bag-of-visual-words matching (section 2), but excluding images at locations closer than  $d_{min}$  meters from the location of  $I$  to ensure that retrieved images do not contain the same scene. A local confusion

score  $\rho$  is then measured over the image  $I$  in a sliding window manner on a dense grid of locations. For a window  $w$  at a particular image position we determine the score as

$$\rho_w = \frac{1}{n} \sum_{k=1}^n \frac{M_w^k}{N_w}, \quad (1)$$

where  $M_w^k$  is the number of tentative feature matches between the window  $w$  and the  $k$ -th “confusing” image, and  $N_w$  is the total number of visual words within the window  $w$ . In other words, the score measures the average number of image matches normalized by the number of detected features in the window. The score is high if a large proportion of visual words (within the window) matches to the set of confusing images and is low in areas with relatively small number of confusing matches. The confusion score can then be used to obtain a segmentation of the image into a layer specific for the particular place (regions with low confusion score) and a confuser layer (regions with high confusion score). In this work we opt for a simple threshold based segmentation, however more advanced segmentation methods respecting image boundaries can be used [35]. The entire process is illustrated in figure 2. Several examples are shown in figure 3. The main parameters of the method are the width  $s$  of the sliding window and the threshold  $t$  on the confusion score. We set  $s = 75$  pixels, where the windows are spaced on a 5 pixel grid in the image, and  $t = 1.5$ , i.e. a window has to have on average 1.5 times more matches than detected features to be deemed confusing. Sensitivity of the place recognition performance to selection of these parameters is evaluated in section 5.

## 4 Place Matching with Confuser Suppression

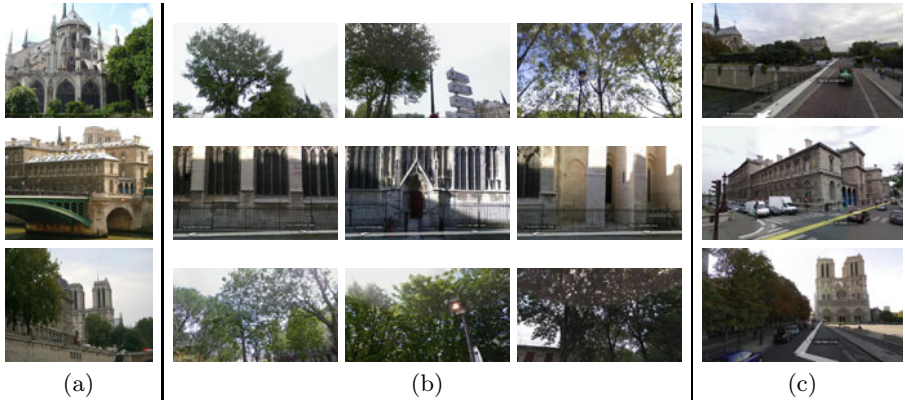
The local confusion score can potentially be used in all stages of the place recognition pipeline, i.e., for vocabulary building, initial retrieval, spatial verification and query expansion. In the following we investigate suppressing confusers in the initial retrieval stage.

To understand the effect of confusers on the retrieval similarity score  $s(\mathbf{q}, \mathbf{v}^i)$  between the query  $\mathbf{q}$  and each database visual word vector  $\mathbf{v}^i$  we can write both the query and the database vector as  $\mathbf{x} = \mathbf{x}_p + \mathbf{x}_c$ , where  $\mathbf{x}_p$  is place specific and  $\mathbf{x}_c$  is due to confusers. The retrieval score is measured by the normalized scalar product (section 2),

$$s(\mathbf{q}, \mathbf{v}^i) = \frac{\mathbf{q}^\top \mathbf{v}^i}{\|\mathbf{q}\| \|\mathbf{v}^i\|} = \frac{(\mathbf{q}_p + \mathbf{q}_c)^\top (\mathbf{v}_p^i + \mathbf{v}_c^i)}{\|\mathbf{q}_p + \mathbf{q}_c\| \|\mathbf{v}_p^i + \mathbf{v}_c^i\|} = \frac{\mathbf{q}_p^\top \mathbf{v}_p^i + \mathbf{q}_c^\top \mathbf{v}_p^i + \mathbf{q}_p^\top \mathbf{v}_c^i + \mathbf{q}_c^\top \mathbf{v}_c^i}{\|\mathbf{q}_p + \mathbf{q}_c\| \|\mathbf{v}_p^i + \mathbf{v}_c^i\|}. \quad (2)$$

If confusers are detected and removed in each database image the terms involving  $\mathbf{v}_c^i$  vanish. Further, if there are no common features between  $\mathbf{q}_c$  and  $\mathbf{v}_p^i$ , i.e. confusers in the query image do not intersect with place specific features in the database,  $\mathbf{q}_c^\top \mathbf{v}_p^i = 0$ . Under these two assumptions, the retrieval score reduces to

$$s(\mathbf{q}, \mathbf{v}^i) = \frac{1}{\|\mathbf{q}_p + \mathbf{q}_c\|} \frac{1}{\|\mathbf{v}_p^i\|} (\mathbf{q}_p^\top \mathbf{v}_p^i) \propto \frac{1}{\|\mathbf{v}_p^i\|} (\mathbf{q}_p^\top \mathbf{v}_p^i) \quad (3)$$



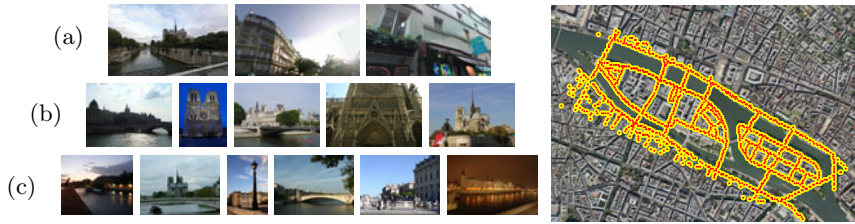
**Fig. 4.** Improvement in place recognition based on suppressing confusing features. (a) The query image. (b) Three top ranked images after initial retrieval and spatial verification. (c) The top ranked image after suppressing confusing image regions. Note that the highly ranked false positive images shown in (b) are suppressed in (c).

For a given query, we are interested only in the ranking of the database images and not the actual value of the score, hence the query normalization dependent on  $\mathbf{q}_c$  can be ignored. This is an interesting property as it suggests that if all confusers are removed from the database, the ranking of database images does not depend on confusers in the query. In practice, however, the second assumption above,  $\mathbf{q}_c^\top \mathbf{v}_p^i = 0$ , might not be always satisfied, since confusers are specific to each place, and not necessary global across the whole database. Hence, some common features between  $\mathbf{q}_c$  and  $\mathbf{v}_p^i$  may remain. Nevertheless, we demonstrate significant improvements in place recognition (section 5) by suppressing confusers on the database side, i.e. setting  $\mathbf{v}_c^i = \mathbf{0}$  for all database images and implicitly exploiting the fact that  $\mathbf{q}_c^\top \mathbf{v}_p^i \ll \mathbf{q}_p^\top \mathbf{v}_p^i$ .

*Implementation:* The local confusion score is pre-computed offline for each image in the database, and all features with a score greater than a certain threshold are suppressed. The remaining features are then indexed using visual words. The initial retrieval, spatial verification and query expansion are performed as outlined in section 2 but for initial retrieval we remove confusing features from the geotagged database. The benefits of suppressing confusing features for place recognition are illustrated in figure 4.

*Discussion:* Note that the proposed confusion score is different from the tf-idf weighting [33], typically used in image retrieval [36,18,13,24], which downweights frequently occurring visual words in the whole database. The tf-idf score is computed independently for each visual word and estimated globally based on the frequency of occurrence of visual words in the whole database, whereas in our case the confusion score is estimated for a local window in each image. The local





**Fig. 5.** Left: examples of (a) geo-tagged images; (b) test query images; (c) non-geo-tagged images. Right: locations of geo-tagged images overlaid on a map of Paris.

confusion score allows removing confusers that are specific to particular images and avoids excessive pruning of features that are confusing in some but hold useful information for other images. Moreover, the top-retrieved images from faraway places, which are used to determine the confusion score, act as place-specific difficult negative “training” examples. This form of supervision is naturally available for georeferenced imagery, but not in the general image retrieval setting. This type of negative supervisory signal is also different from the clean (positive and negative) supervision typically used in feature selection methods in object category recognition [25,26,27] and retrieval [28,29,30]. In our case, obtaining verified positive examples would require expensive image matching, and for many places positive examples are not available due to sparse location sampling of the image database.

## 5 Experimental Evaluation

First, we describe image datasets and the performance measure, which will be used to evaluate the proposed place recognition method. In the following subsection, we test the sensitivity to key parameters and present place recognition results after different stages of the algorithm.

### 5.1 Image Datasets

*Geotagged google street-view images:* The geotagged dataset consists of about 17K images automatically downloaded from Google StreetView [1]. We have downloaded all available images in a district of Paris covering roughly an area of  $1.7 \times 0.5$  kilometers. The full  $360 \times 180$  panorama available at each distinct location is represented by 12 perspective images with resolution  $936 \times 537$  pixels. Example images are shown in figure 5.1(a) and image locations overlaid on a map are shown in figure 5.1(right).

*Non-geotagged images:* Using keyword and location search we have downloaded about 8K images from the photo-sharing website Panoramio [37]. Images were downloaded from roughly the same area as covered by the geotagged database.

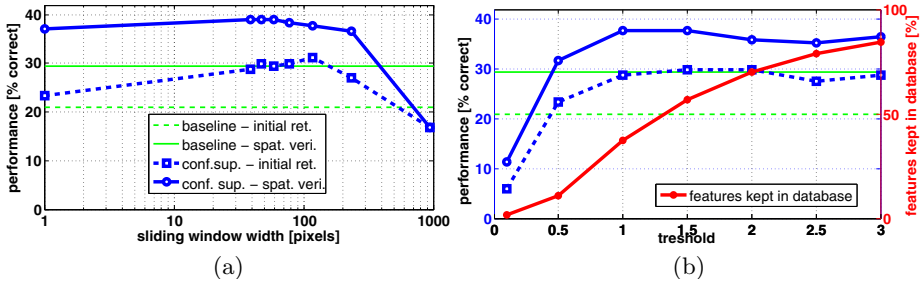
The location information on photo-sharing websites is very coarse and noisy and therefore some images are from other parts of Paris or even different cities. Apart from choosing which images to download, we do not use the location information in any stage of our algorithm and treat the images as non-geotagged.

*Test set:* In addition, a test set of 200 images was randomly sampled from the non-geotagged image data. These images are set aside as unseen query images and are not used in any stage of the processing apart from testing. Examples of query images and non-geotagged images are shown in figure 5.1 (b) and (c).

*Performance measures:* Given a test query image the goal is to recognize the place by finding an image from the geotagged database depicting the same place, i.e., the same 3D structure. We measure the recognition performance by the number of test images (out of 200 test queries), for which the top-ranked image from the geotagged database correctly depicts the same place. The ground truth is obtained manually by inspection of the visual correspondence between the query and the top retrieved image. The overall performance is then measured by the percentage of correctly matched test images. As 33 images (out of the 200 randomly sampled queries) do not depict places within the geotagged database, the perfect score of 100% would be achieved when the remaining 167 images are correctly matched.

## 5.2 Performance Evaluation

*Parameter settings:* We have found that parameter settings of the baseline place recognition, such as the vocabulary size  $K$  ( $=10^5$ ), the top  $m$  ( $=50$ ) candidates for spatial verification or the minimum number of inliers (20) to deem a successful match work well with confuser suppression and keep them unchanged throughout the experimental evaluation. For confuser suppression, we set the minimal spatial distance to obtain confusing images to one fifth of the map (about 370 meters) and consider the top  $n = 20$  confusing images. In the following, we evaluate sensitivity of place recognition to the sliding window width,  $s$ , and confuser score threshold,  $t$ . We explore two one-dimensional slices of the 2-D parameter space, by varying  $s$  for fixed  $t = 1.5$ , figure 6(a)), and varying  $t$  for fixed  $s = 75$  pixels, (figure 6(b)). From graph 6(a), we note that a good performance is obtained for window sizes between 30 and 100 pixels. The window size specially affects the performance of the initial bag-of-visual-words matching and less so the results after spatial verification. This may be attributed to a certain level of spatial consistency implemented by the intermediate-size windows, where groups of spatially-localized confusing features are removed. However, even removing individual features ( $s=1$  pixel) enables retrieving many images, initially low-ranked by the baseline approach, within the top 50 matches so that they are later correctly re-ranked with spatial verification. Graph 6(b) again shows good place recognition performance over a wide range of confuser detection thresholds. The chosen value  $t = 1.5$  represents a good compromise between the database size and place recognition performance, keeping around 60% of originally detected



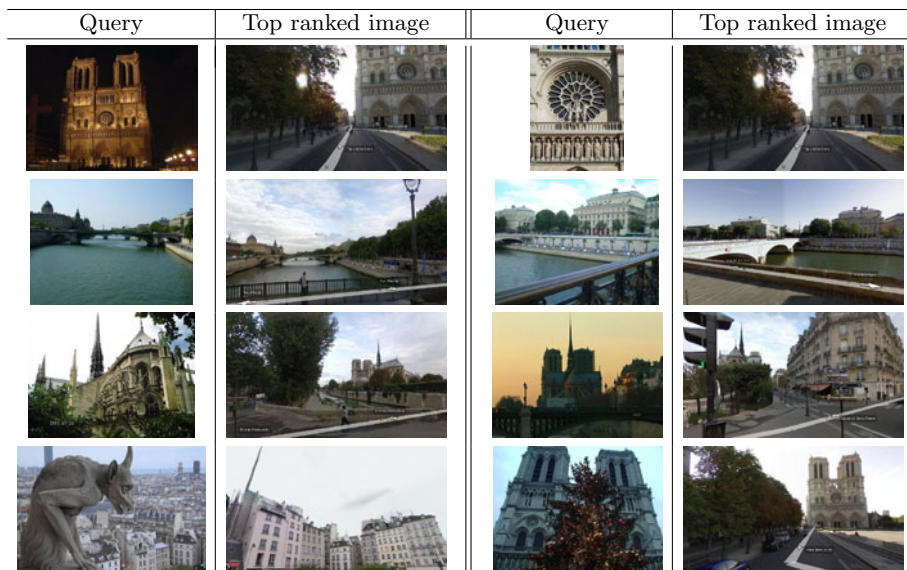
**Fig. 6.** (a) Place recognition performance for varying confuser sliding window width  $s$ . (b) Place recognition performance (left axis) and percentage of features kept in the geotagged database (right axis) for varying confuser detection threshold  $t$ .

**Table 1.** Percentage of correctly localized test queries for different place recognition approaches

Method	% correct <i>initial retrieval</i>	% correct <i>with spatial verification</i>
a. Baseline place recognition	20.96	29.34
b. Query expansion	26.35	41.92
c. Confuser suppression	29.94	37.72
d. Confuser suppression+Query expansion	32.93	<b>47.90</b>

features. However, with a small loss in initial retrieval performance, even a lower threshold  $t = 1$  can be potentially used.

*Overall place recognition performance:* In the remainder of this section, we evaluate the overall place recognition performance after each stage of the proposed method. Results are summarized in table 1. It is clear that spatial re-ranking improves initial bag-of-visual-words matching in all stages of the proposed algorithm. This illustrates that the initial bag-of-visual-words matching can be noisy and does not always return the correct match at the top rank, however, correct matches can be often found within the top 50 best matches. Both the query expansion and non-informative feature suppression also significantly improve place recognition performance of the baseline approach. When applied together, the improvement is even bigger correctly recognizing 47.90% of places in comparison with only 41.92% using query expansion alone and 37.72% using confuser suppression alone. This could be attributed to the complementarity of both methods. The place query expansion improves recall by enhancing the query using relevant features found in the non-geotagged database, whereas confuser suppression removes confusing features responsible for many highly ranked false positives. Overall, the performance with respect to the baseline bag-of-visual-words method (without spatial re-ranking) is more than doubled from 20.96% to 47.90% correctly recognized place queries – a significant improvement on the challenging real-world test set. Examples of correct place recognition results are



**Fig. 7.** Examples of correct place recognition results. Each image pair shows the query image (left) and the best match from the geotagged database (right). Note that query places are recognized despite significant changes in viewpoint (bottom left), lighting conditions (top left), or presence of large amounts of clutter and occlusion (bottom right).



**Fig. 8.** Examples of challenging test query images, which were not found in the geotagged database

shown in figure 7. Examples of non-localized test queries are shown in figure 8. Many of the non-localized images represent very challenging examples for current matching methods due to large changes in viewpoint, scale and lighting conditions. It should be also noted that the success of query expansion depends on the availability of additional photos for a particular place. Places with additional images have a higher chance to be recognized.

## 6 Conclusions

We have demonstrated that place recognition performance for challenging real-world query images can be significantly improved by automatic detection and suppression of spatially localized groups of confusing non-informative features

in the geotagged database. Confusing features are found by matching places spatially far on the map – a negative supervisory signal readily available in geotagged databases. We have also experimentally demonstrated that the method combines well with the state of the art bag-of-features model and query expansion.

Detection of spatially defined confusing image regions opens up the possibility of their automatic clustering and category-level analysis (when confusers correspond to trees, pavement or buses), determining their geospatial scale (trees might appear everywhere, whereas a particular type of buses may not), and reasoning about their occurrence in conjunction with location-specific objects (a tree in front of a house may still be a characteristic feature). Next, we plan to include such category-level place analysis in the current framework to further improve the place recognition performance.

*Acknowledgements.* We are grateful for financial support from the MSR-INRIA laboratory, ANR-07-BLAN-0331-01, FP7-SPACE-241523 PRoViScout and MSM6840770038.

## References

1. <http://maps.google.com/help/maps/streetview/>
2. <http://www.bing.com/maps/>
3. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR (2007)
4. Aguera y Arcas, B.: Augmented reality using Bing maps. Talk at TED (2010)
5. Quack, T., Leibe, B., Van Gool, L.: World-scale mining of objects and events from community photo collections. In: CIVR (2008)
6. Li, Y., Crandall, D., Huttenlocher, D.: Landmark classification in large-scale image collections. In: ICCV (2009)
7. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH (2006)
8. Havlena, M., Torii, A., Pajdla, T.: Efficient structure from motion by graph optimization. In: ECCV (2010)
9. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or “How do I organize my holiday snaps?” In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 414–431. Springer, Heidelberg (2002)
10. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: WS-SLCV, ECCV (2004)
11. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
12. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
14. Shao, H., Svoboda, T., Tuytelaars, T., van Gool, L.: Hpat indexing for fast object/scene recognition based on local appearance. In: CIVR (2003)

15. Silpa-Anan, C., Hartley, R.: Localization using an image-map. In: ACRA (2004)
16. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT (2006)
17. Cummins, M., Newman, P.: Highly scalable appearance-only SLAM - FAB-MAP 2.0. In: Proceedings of Robotics: Science and Systems, Seattle, USA (2009)
18. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
19. Hays, J., Efros, A.: im2gps: estimating geographic information from a single image. In: CVPR (2008)
20. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: CVPR (2009)
21. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.-M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 427–440. Springer, Heidelberg (2008)
22. Simon, I., Snavely, N., Seitz, S.: Scene summarization for online image collections. In: SIGGRAPH (2006)
23. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large-scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
24. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problem. In: WS-LAVD, ICCV (2009)
25. Lee, Y., Grauman, K.: Foreground focus: Unsupervised learning from partially matching images. IJCV 85 (2009)
26. Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
27. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. IEEE PAMI 29 (2007)
28. Kulis, B., Jain, P., Grauman, K.: Fast similarity search for learned metrics. IEEE PAMI 31 (2009)
29. Torresani, L., Szummer, M., Fitzgibbon, A.: Learning query-dependent prefilters for scalable image retrieval. In: CVPR (2009)
30. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
31. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
32. Muja, M., Lowe, D.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP (2009)
33. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24 (1988)
34. Chum, O., Matas, J., Obdrzalek, S.: Enhancing RANSAC by generalized model optimization. In: ACCV (2004)
35. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV (2001)
36. Jegou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR (2009)
37. <http://www.panoramio.com/>

# Semantic Label Sharing for Learning with Many Categories

Rob Fergus<sup>1</sup>, Hector Bernal<sup>2</sup>, Yair Weiss<sup>3</sup>, and Antonio Torralba<sup>2</sup>

<sup>1</sup> Courant Institute, New York University  
fergus@cs.nyu.edu

<sup>2</sup> CSAIL, MIT

{hectorbernal, torralba}@csail.mit.edu

<sup>3</sup> School of Computer Science

Hebrew University

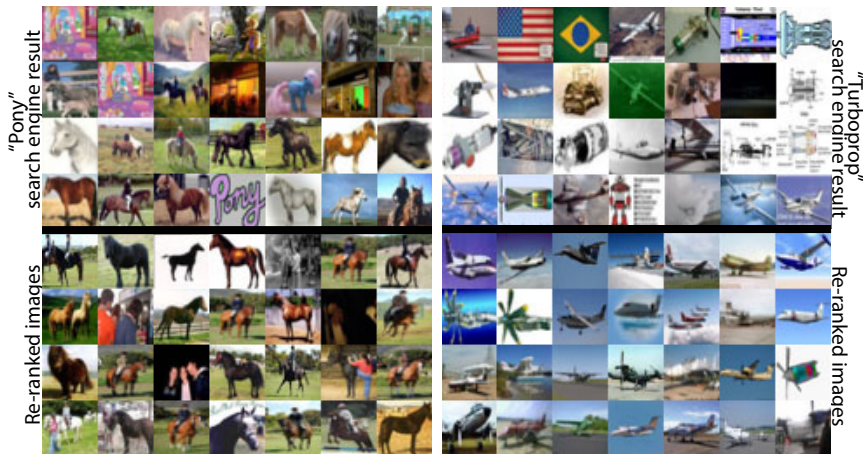
yweiss@cs.huji.ac.il

**Abstract.** In an object recognition scenario with tens of thousands of categories, even a small number of labels per category leads to a very large number of total labels required. We propose a simple method of *label sharing* between semantically similar categories. We leverage the WordNet hierarchy to define semantic distance between any two categories and use this semantic distance to share labels. Our approach can be used with any classifier. Experimental results on a range of datasets, upto 80 million images and 75,000 categories in size, show that despite the simplicity of the approach, it leads to significant improvements in performance.

## 1 Introduction

Large image collections on the Internet and elsewhere contain a multitude of scenes and objects. Recent work in computer vision has explored the problems of visual search and recognition in this challenging environment. However, all approaches require some amount of hand-labeled training data in order to build effective models. Working with large numbers of images creates two challenges: first, labeling a representative set of images and, second, developing efficient algorithms that scale to very large databases.

Labeling Internet imagery is challenging in two respects: first, the sheer number of images means that the labels will only ever cover a small fraction of images. Recent collaborative labeling efforts such as Peekaboom, LabelMe, ImageNet [2,3,4] have gathered millions of labels at the image and object level. However this is but a tiny fraction of the estimated 10 billion images on Facebook, let alone the hundreds of petabytes of video on YouTube. Second, the diversity of the data means that many thousands of classes will be needed to give an accurate description of the visual content. Current recognition datasets use 10's to 100's of classes which give a hopelessly coarse quantization of images into discrete categories. The richness of our visual world is reflected by the enormous number of nouns present in our language: English has around 70,000 that



**Fig. 1.** Two examples of images from the Tiny Images database [1] being re-ranked by our approach, according to the probability of belonging to the categories “pony” and “turboprop” respectively. *No training labels were available for either class.* However 64,185 images from the total of 80 million were labeled, spread over 386 classes, some of which are semantically close to the two categories. Using these labels in our semantic label sharing scheme, we can dramatically improve search quality.

correspond to actual objects [5]. This figure loosely agrees with the 30,000 visual concepts estimated by psychologists [6]. Furthermore, having a huge number of classes dilutes the available labels, meaning that, on average, there will be relatively few annotated examples per class (and many classes might not have any annotated data).

To illustrate the challenge of obtaining high quality labels in the scenario of many categories, consider the CIFAR-10 dataset constructed by Alex Krizhevsky and Geoff Hinton [7]. This dataset provides human labels for a subset of the Tiny Images [1] dataset which was obtained by querying Internet search engines with over 70,000 search terms. To construct the labels, Krizhevsky and Hinton chose 10 classes “airplane”, “automobile”, “bird”, “cat”, “deer”, “dog”, “frog”, “horse”, “ship”, “truck”, and for each class they used the WordNet hierarchy to construct a set of hyponyms. The labelers were asked to examine all the images which were found with a search term that is a hyponym of the class. As an example, some of the hyponyms of ship are “cargo ship”, “ocean liner”, and “frigate”. The labelers were instructed to reject images which did not belong to their assigned class. Using this procedure, labels on a total of 386 categories (hyponyms of the 10 classes listed above) were collected at a cost of thousands of dollars.

Despite the high cost of obtaining these labels, the 386 categories are of course a tiny subset of the possible labels in the English language. Consider for example the words “pony” and “turboprop” (Fig. 1). Neither of these is considered a



hyponym of the 10 classes mentioned above. Yet there is obvious information in the labeled data for “horse” and “airplane” that we would like to use to improve the search engine results of “pony” and “turboprop”.

In this paper, we provide a very simple method for sharing labels between categories. Our approach is based on a basic assumption – we expect the classifier output for a single category to degrade gracefully with semantic distance. In other words, although horses are not exactly ponies, we expect a classifier for “pony” to give higher values for “horses” than to “airplanes”. Our scheme, which we call “Semantic Label Sharing” gives the performance shown in Fig. 1. Even though we have no labels for “pony” and “turboprop” specifically, we can significantly improve the performance of search engines by using label sharing.

## 1.1 Related Work

Various recognition approaches have been applied to Internet data, with the aim of re-ranking, or refining the output of image search engines. These include: Li *et al.* [8], Fergus *et al.* [9], Berg *et al.* [10], amongst others. Our approach differs in two respects: (i) these approaches treat each class independently; (ii) they are not designed to scale to the billions of images on the web.

Sharing information across classes is a widely explored concept in vision and learning, and takes many different forms. Some of the first approaches applied to object recognition are based on neural networks in which sharing is achieved via the hidden layers which are common across all tasks [11,12]. Error correcting output codes [13] also look at a way of combining multi-class classifiers to obtain better performance. Another set of approaches tries to transfer information from one class to another by regularizing the parameters of the classifiers across classes. Torralba *et al.*, Opelt *et al.* [14,15] demonstrated its power in sharing useful features between classes within a boosting framework. Other approaches transfer information across object categories by sharing a common set of parts [16,17], by sharing transformations across different instances [18,19,20], or by sharing a set of prototypes [21]. Common to all those approaches is that the experiments are always performed with relatively few classes. Furthermore, it is not clear how these techniques would scale to very large databases with thousands of classes.

Our sharing takes a different form to these approaches, in that we impose sharing on the class labels themselves, rather than in the features or parameters of the model. As such, our approach has the advantage that it is independent of the choice of the classifier.

## 2 Semantic Label Sharing

Following [22] we define the semantic distance between two classes using a tree defined by WordNet<sup>1</sup>. We use a simple metric that measures the intersection between the ancestors of two words: the semantic distance  $S_{ij}$  between classes  $i$  and

<sup>1</sup> Wordnet is graph-structured and we convert it into a tree by taking the most common sense of a word.

$j$  (which are nodes in the tree) is defined as the number of nodes shared by their two parent branches, divided by the length of the longest of the two branches, i.e.  $S_{ij} = \text{intersect}(\text{par}(i), \text{par}(j)) / \max(\text{length}(\text{par}(i)), \text{length}(\text{par}(j)))$ , where  $\text{par}(i)$  is the path from the root node to node  $i$ . For instance, the semantic similarity between a “felis domesticus” and “tabby cat” is 0.93, while the distance between “felis domesticus” and a “tractor trailer” is 0.21. We construct a sparse semantic affinity matrix  $A = \exp(-\kappa(1 - S))$ , with  $\kappa = 10$  for all the experiments in this paper. For the class “airbus”, the nearest semantic classes are: “airliner” (0.49), “monoplane” (0.24), “dive bomber” (0.24), “twinjet” (0.24), “jumbo jet” (0.24), and “boat” (0.03). A visualization of  $A$  and a closeup are shown in Fig. 3(a) and (b).

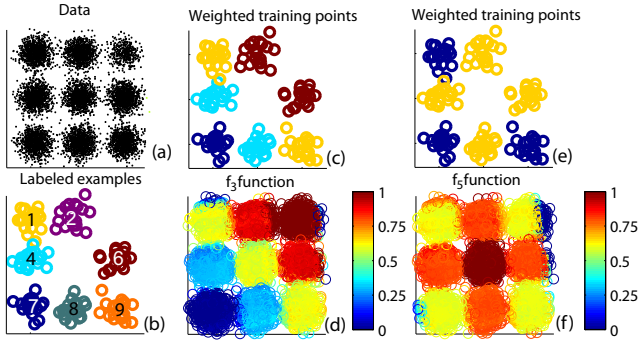
Let us assume we have a total of  $C$  classes, hence  $A$  will be a  $C \times C$  symmetric matrix. We are given  $L$  labeled examples in total, distributed over these  $C$  classes. The labels for class  $c$  are represented by a binary vector  $y_c$  of length  $L$  which has values 1 for positive hand-labeled examples and 0 otherwise. Hence positive examples for class  $c$  are regarded as negative labels for all other classes.  $Y = \{y_1, \dots, y_C\}$  is an  $N \times C$  matrix holding the label vectors from all classes.

We share labels between classes by replacing  $Y$  with  $YA$ . This simple operation has a number of effects:

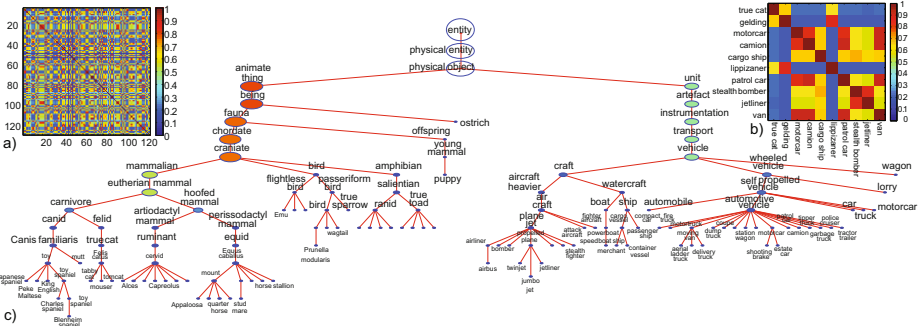
- Positive examples are copied between classes, weighted according to their semantic affinity. For example, the label vector for “felis domesticus” previously had zero values for the images of “tabby cat”, but now these elements are replaced by the value 0.93.
- However, labels from unrelated classes will only deviate slightly from their original state of 0 (dependent on the value of  $\kappa$ ).
- Negative labeled examples from classes outside the set of  $C$  are unaffected by  $A$  (since they are 0 across all rows of  $Y$ ).
- Even if each class has only a few labeled examples, the multiplication by  $A$  will effectively pool examples across semantically similar classes, dramatically increasing the number that can be used for training, provided semantically similar classes are present amongst the set of  $C$ .

The effect of this operation is illustrated in two examples on toy data, shown in Fig. 2. These examples show good classifiers can be trained by sharing labels between classes, given knowledge of the inter-class affinities, even when no labels are given for the target class. In Fig. 2, there are 9 classes but label data is only given for 7 classes. In addition to the labels, the system also has access to the affinities among the 9 classes. This information is enough to build classification functions for the classes with no labels (Fig. 2(d) and (f)).

From another perspective, our sharing mechanism turns the original classification problem into a regression problem: the formerly binary labels in  $Y$  become real-values in  $YA$ . As such we can adapt many types of classifiers to minimize regression error rather than classification error.



**Fig. 2.** Toy data illustrating our sharing mechanism between 9 different classes (a) in discrete clusters. For 7 of the 9 classes, a few examples are labeled (b). No labels exist for the classes 3 and 5. (c): Labels re-weighted by affinity to class 3. (Red=high affinity, Blue=low affinity). (d): This plot shows the semi-supervised learning solution  $f_{\text{class}=3}$  using weighted labels from (c). The value of the function  $f_{\text{class}=3}$  on each sample from (a) is color coded. Dark red corresponds to the samples more likely to belong to class 3. (e): Labels re-weighted by affinity to class 5. (d): Solution of semi-supervised learning solution  $f_{\text{class}=5}$  using weighted labels from (e).



**Fig. 3.** Wordnet sub-tree for a subset of 386 classes used in our experiments. The associated semantic affinity matrix  $A$  is shown in (a), along with a closeup of 10 randomly chosen rows and columns in (b).

### 3 Sharing in Semi-supervised Learning

Semi-supervised learning is an attractive option in settings where very few training examples exist since the density of the data can be used to regularize the solution. This can help prevent over-fitting the few training examples and yield superior solutions. A popular class of semi-supervised algorithms are based on the graph Laplacian and we use an approach of this type.

We briefly describe semi-supervised learning in a graph setting. In addition to the  $L$  labeled examples  $(X_L, Y_L) = \{(x_1, y_1), \dots, (x_L, y_L)\}$  introduced above, we have an additional  $U$  unlabeled images  $X_u = \{x_{L+1}, \dots, x_N\}$ , for a total

of  $N$  images. We form a graph where the vertices are the images  $X$  and the edges are represented by an  $N \times N$  matrix  $W$ . The edge weighting is given by  $W_{ij} = \exp(-\|x_i - x_j\|^2/2\epsilon^2)$ , the visual affinity between images  $i$  and  $j$ . Defining  $D = \text{diag}(\sum_j W_{ij})$ , we define the normalized graph Laplacian to be:  $L = I - D^{-1/2}WD^{-1/2}$ . We use  $L$  to measure the smoothness of solutions over the data points, desiring solutions that agree with the labels but are also smooth with respect to the graph. In the single class case we want to minimize:

$$J(f) = f^T Lf + \sum_{i=1}^l \lambda(f_i - y_i)^2 = f^T Lf + (f - y)^T \Lambda(f - y) \quad (1)$$

where  $\Lambda$  is a diagonal matrix whose diagonal elements are  $\Lambda_{ii} = \lambda$  if  $i$  is a labeled point and  $\Lambda_{ii} = 0$  for unlabeled points. The solution is given by solving the  $N \times N$  linear system  $(L + \Lambda)f = \Lambda y$ .

This system is impractical to solve for large  $N$ , thus it is common [23,24,25] to reduce the dimension of the problem by using the smallest  $k$  eigenvectors of  $L$  (which will be the smoothest)  $U$  as a basis with coefficients  $\alpha$ :  $f = U\alpha$ . Substituting into Eqn. 1, we find the optimal coefficients  $\alpha$  to be the solution of the following  $k \times k$  system:

$$(\Sigma + U^T \Lambda U)\alpha = U^T \Lambda y \quad (2)$$

where  $\Sigma$  is a diagonal matrix of the smallest  $k$  eigenvectors of  $L$ . While this system is easy to solve, the difficulty is computing the eigenvectors an  $O(N^2)$  operation.

Fergus *et al.* [26] introduced an efficient scheme for computing approximate eigenvectors in  $O(N)$  time. This approach proceeds by first computing numerical approximations to the eigenfunctions (the limit of the eigenvectors as  $N \rightarrow \infty$ ). Then approximations to the eigenvectors are computed via a series of 1D interpolations into the numerical eigenfunctions. The resulting approximate eigenvectors (and associated eigenvalues) can be used in place of  $U$  and  $\Sigma$  in Eqn. 2.

Extending the above formulations to the multi-class scenario is straightforward. In a multi-class problem, the labels will be held in an  $N \times C$  binary matrix  $Y$ , replacing  $y$  in Eqn. 2. We then solve for the  $N \times C$  matrix  $F$  using the approach of Fergus *et al.* Utilizing the semantic sharing from Section 2 is simple, with  $Y$  being replaced with  $YA$ .

## 4 Experiments

We evaluate our sharing framework on two tasks: (a) improving the performance of images returned by Internet search engines; (b) object classification. Note that the first problem consists of a set of 2-class problems (e.g. sort the pony images from the non-pony images), while the second problem is a multi-class classification with many classes.

These tasks are performed on three datasets linked to the Tiny Images database [1], a diverse and highly variable image collection downloaded from the Internet:

- *CIFAR*: This consists of 63,000 images from 126 classes selected<sup>2</sup> from the CIFAR-10 dataset [7], which is a hand-labeled sub-set of the Tiny Images. These keywords and their semantic relationship to one another are shown in Fig. 3. For each keyword, we randomly choose a fixed test-set of 75 positive and 150 negative examples, reflecting the typical signal-to-noise ratio found in images from Internet search engines. From the remaining images for each class, we randomly draw a validation set of 25/50 +ve/-ve examples. The training examples consist of +ve/-ve pairs drawn from the remaining pool of 100 positive/negative images for each keyword.
- *Tiny*: The whole Tiny Images dataset, consisting of 79,302,017 images distributed over 74,569 classes (keywords used to download the images from the Internet). No human-provided labels are available for this dataset, thus instead we use the noisy labels from the image search engines. For each class we assume the first 5 images to be true positive examples. Thus over the dataset, we have a total of 372,845 (noisy) positive training examples, and the same number of negative examples (drawn at random). For evaluation, we can use labeled examples from either the *CIFAR* or *High-res* datasets.
- *High-res*: This is a sub-set of 10,957,654 images from the Tiny Images, for which the high-resolution original image exists. These images span 53,564 different classes, distributed evenly over all classes within the Tiny Images dataset. As with the *Tiny* dataset, we use no hand-labeled examples for training, instead using the first 5 examples for each class as positive examples (and 5 negative drawn randomly). For evaluation, we use 5,357 human-labeled images split into 2,569 and 2,788 positive and negative examples of each class respectively.

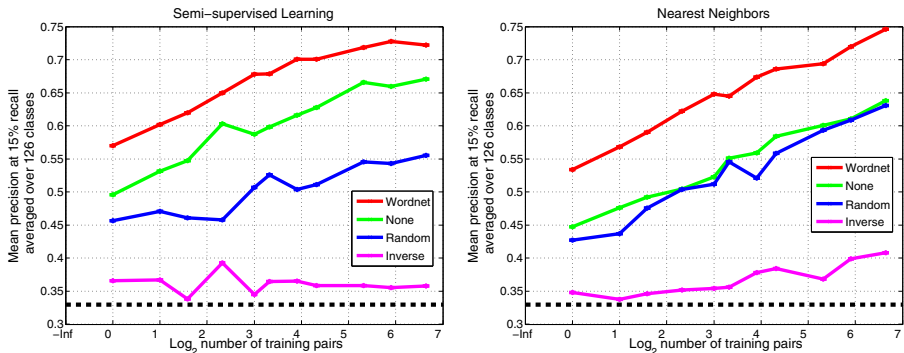
**Pre-processing:** For all datasets, each image is represented by a single Gist descriptor. In the case of the *Tiny* and *CIFAR* datasets, a 384-D descriptor is used which is then mapped down to 32 and 64 dimensions using PCA, for *Tiny* and *CIFAR* respectively. For the *High-res* dataset, a 512-D Gist descriptor is mapped down to 48-D using PCA.

#### 4.1 Re-ranking Experiments

On the re-ranking task we first use the *CIFAR* dataset to quantify the effects of semantic sharing. For each class separately we train a classifier on the training set (possibly using sharing) and use it to re-rank the 250 test images, measuring the precision at 15% recall. Unless otherwise stated, the classifier used is the semi-supervised approach of Fergus *et al.* [26].

In Fig. 4(left) we explore the effects of semantic sharing, averaging performance over all 126 classes. The validation set is used to automatically select the optimal values of  $\kappa$  and  $\lambda$ . The application of the Wordnet semantic affinity matrix can be seen to help performance. If the semantic matrix is randomly permuted (but with the diagonal fixed to be 1), then this is somewhat worse

<sup>2</sup> The selected classes were those that had at least 200 positive labels and 300 negative labels, to enable accurate evaluation.



**Fig. 4.** *Left:* Performance for different sharing strategies with the semi-supervised learning approach of [26] as the number of training examples is increased, using 126 classes in the *CIFAR* dataset. *Right:* As for (left) but with a nearest neighbor classifier. The black dashed line indicates chance level performance. When the Wordnet matrix is used for sharing it gives a clear performance improvement (red) to both methods over no sharing [26] (green). However, if the semantic matrix does not reflect the similarity between classes, then it hinders performance (e.g. random (blue) and inverse (magenta) curves).

than not using sharing. But if the sharing is inverted (by replacing  $A$  with  $1 - A$  and setting the diagonal to 1), it clearly hinders performance. The same pattern of results can be seen in Fig. 4(right) for a nearest neighbor classifier. Hence the semantic matrix must reflect the relationship between classes if it is to be effective. In Fig. 5 we show examples of the re-ranking, using the semi-supervised learning scheme in conjunction with the Wordnet affinity matrix.

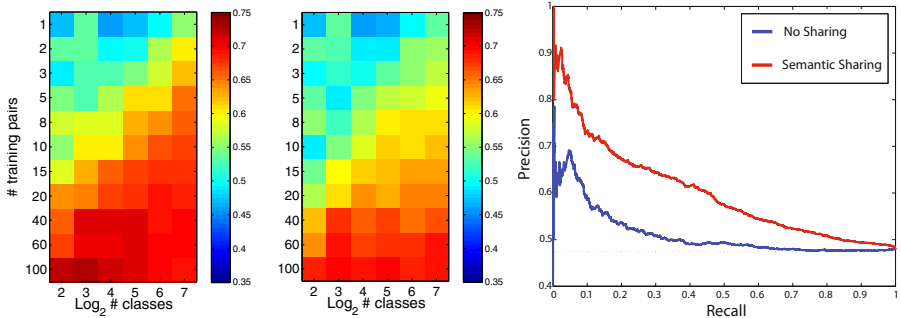
In Fig. 6(left & middle), we perform a more systematic exploration of the effects of Wordnet sharing. For these experiments we use fixed values of  $\kappa = 5$  and  $\lambda = 1000$ . Both the number of classes and number of images are varied, and the performance recorded with and without the semantic affinity matrix. The sharing gives a significant performance boost, particularly when few training examples are available.

The sharing behavior can be used to effectively learn classes for which we have zero training examples. In Fig. 7, we explore what happens when we allocate 0 training images to one particular class (the left-out class) from the set of 126, while using 100 training pairs for the remaining 125 classes. When the sharing matrix is not used, the performance of the left-out class drops significantly, relative to its performance when training data is available (i.e. the point for each left-out class falls below the diagonal). But when sharing is used, the drop in performance is relatively small, with points being spread around the diagonal.

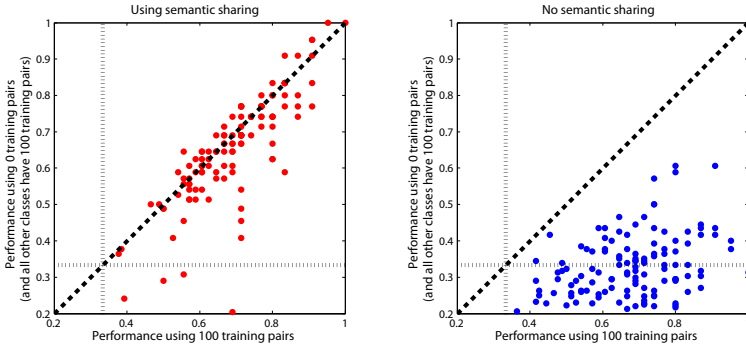
Motivated by Fig. 7, we show in Fig. 8 the approach applied to the *Tiny* dataset, using the human-provided labels from the *CIFAR* dataset. However, no *CIFAR* labels exist for the two classes selected (Pony, Turboprop). Instead, we used the Wordnet matrix to share labels from semantically similar classes for



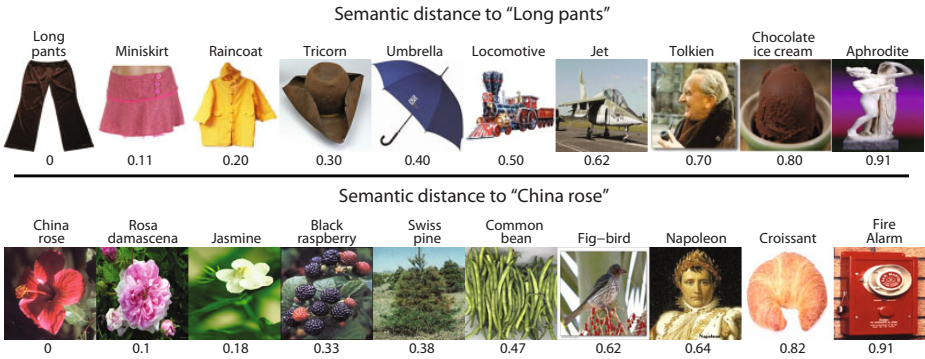
**Fig. 5.** Test images from 7 keywords drawn from the 126 class *CIFAR* dataset. The border of each image indicates its label (used for evaluation purposes only) with respect to the keyword, green = +ve, red = -ve. The top row shows the initial ranking of the data, while the bottom row shows the re-ranking of our approach trained on 126 classes with 100 training pairs/classes.



**Fig. 6.** *Left & Middle:* The variation in precision for the semi-supervised approach as the number of training examples is increased, using 126 classes with (left) and without (middle) Wordnet sharing. Note the improvement in performance for small numbers of training examples when the Wordnet sharing matrix is used. *Right:* Evaluation of our sharing scheme for the re-ranking task on the 10 million image *High-res* dataset, using 5,357 test examples. Our classifier was trained using 0 hand-labeled examples and 5 noisy labels per class. Using a Wordnet semantic affinity matrix over the 53,564 classes gives a clear boost to performance.



**Fig. 7.** An exploration of the performance with 0 training examples for a single class, if all the other classes have 100 training pairs. *Left:* By using the sharing matrix  $A$ , we can obtain a good performance by transferring labels from semantically similar classes. *Right:* Without it, the performance drops significantly.



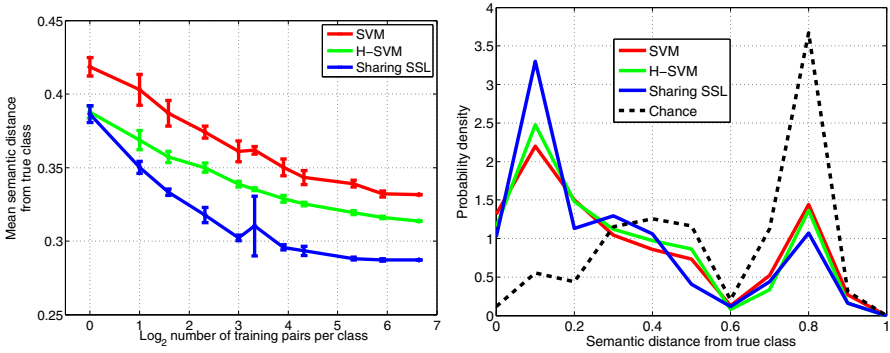
**Fig. 8.** Our semantic distance performance metric for two examples “Long pants” and “China rose”. The other images are labeled with their semantic distance to the two examples. Distances under 0.2 correspond to visual similar objects.

which labels do exist. The qualitatively good results demonstrated in Fig. 4 can only be obtained relatively close to the 126 keywords for which we have labels.

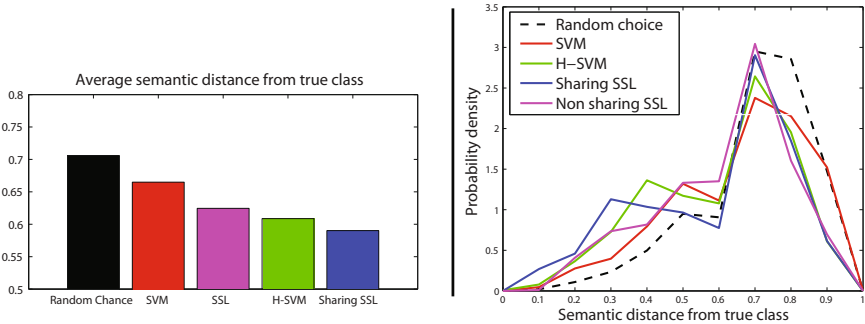
This performance gain obtained by Wordnet sharing is quantified in a large-scale setting in Fig. 6 (right) using the *High-res* dataset. Chance level performance corresponds to  $2569 / (2569 + 2788) = 48\%$ . Without any sharing, the semi-supervised scheme (blue) gives a modest performance. But when the Wordnet sharing is added, there is significant performance boost.

Our final re-ranking experiment applies the semantic sharing scheme to the whole of the *Tiny* dataset (with no CIFAR labels used). With 74,569 classes, many will be very similar visually and our sharing scheme can be expected to greatly assist performance. In Fig. 11 we show qualitative results for 4 classes. The semi-supervised algorithm takes around 0.1 seconds to perform each re-ranking (since the eigenfunctions are precomputed), compared to over 1 minute



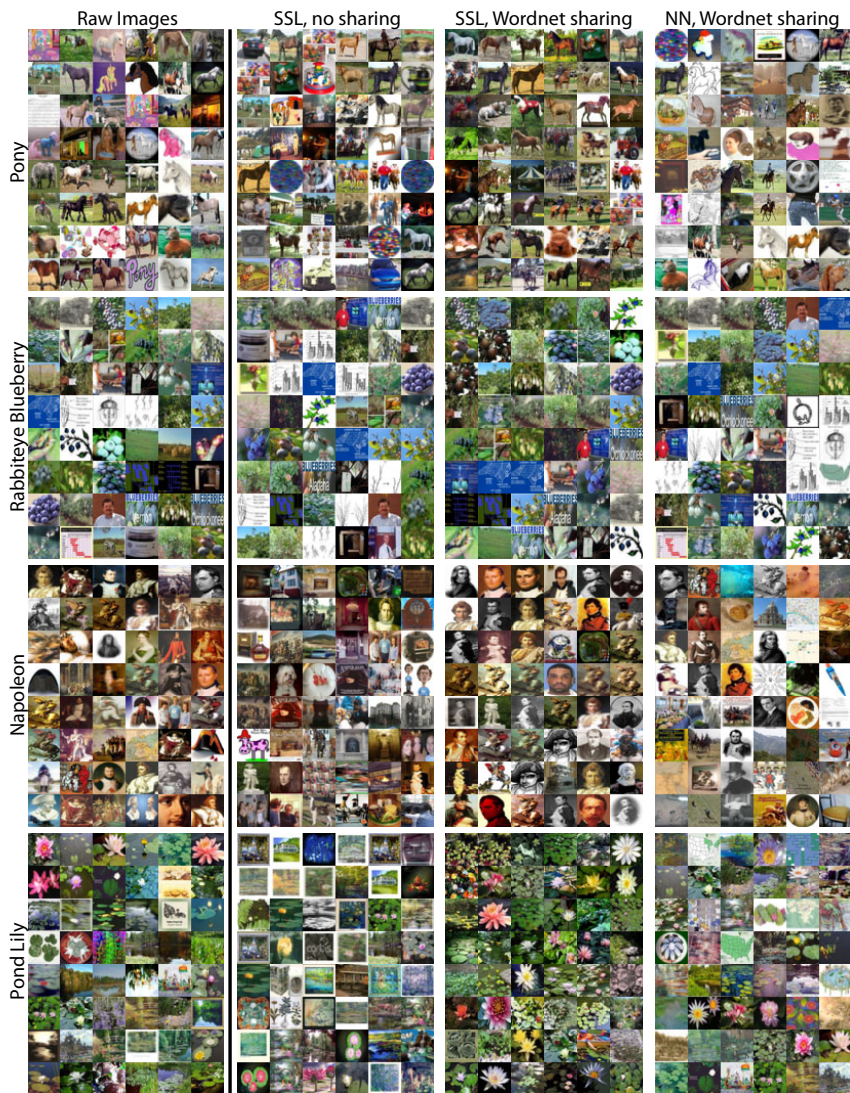


**Fig. 9.** Comparison of approaches for classification on the *CIFAR* dataset. Red: 1 vs all linear SVM; Green: Hierarchical SVM approach of Marszalek and Schmid [27]; Blue: Our semantic sharing scheme in the semi-supervised approach of [26]; Black: Chance. *Left:* Mean semantic distance of test examples to true class as the number of labeled training examples increases (smaller is better). *Right:* For 100 training examples per class, the distribution of distances for the positive test examples. Our sharing approach has a significantly lower mean semantic distance, with a large mass at a distance < 0.2, corresponding to superior classification performance. See Fig. 8 for an illustration of semantic distance.



**Fig. 10.** Comparison of approaches for classification on the *High-res* dataset. Red: 1 vs all linear SVM; Green: Hierarchical SVM approach of Marszalek and Schmid [27]; Magenta: the semi-supervised scheme of [26]; Blue: [26] with our semantic sharing scheme; Black: Random chance. *Left:* Bar chart showing mean semantic distance from true label on test set. *Right:* The distribution of distances for each method on the test set. Our approach has more mass at a distance < 0.2, indicating superior performance.

for the nearest-neighbor classifier. These figures show qualitatively that the semi-supervised learning scheme with semantic sharing clearly improves search performance over the original ranking and that without the sharing matrix the performance drops significantly.



**Fig. 11.** Sample results of our semantic label sharing scheme on the *Tiny* dataset (79 million images). 0 hand-labeled training examples were used. Instead, the first 5 images of each of the 74,569 classes were taken as positive examples. Using these labels, classifiers were trained for 4 different query classes: “pony”, “rabbiteye blueberry”, “Napoleon” and “pond lily”. Column 1: the raw image ranking from the Internet search engine. Column 2: re-ranking using the semi-supervised scheme without semantic sharing. Column 3: re-ranking with semi-supervised scheme and semantic sharing. Column 4: re-ranking with a nearest-neighbor classifier and semantic sharing. Without semantic sharing, the classifier only has 5 positive training examples, thus performs poorly. But with semantic sharing it can leverage the semantically close examples from the pool of  $5 \times 74,569 = 372,845$  positive examples.

## 4.2 Classification Experiments

Classification with many classes is extremely challenging. For example, picking the correct class out of 75,000 is something that even humans typically cannot do. Hence instead of using standard metrics, we measure how far the predicted class is from the true class, as given by the semantic distance matrix  $S$ . Under this measure the true class has distance 0, while 1 indicates total dissimilarity. Fig. 8 illustrates this metric with two example images and a set of samples varying in distance from them.

We compare our semantic sharing approach in the semi-supervised learning framework of [26] to two other approaches: (i) linear 1-vs-all SVM; (ii) the hierarchical SVM approach of Marszalek and Schmid [27]. The latter method uses the semantic relationships between classes to construct a hierarchy of SVMs. In implementing this approach, we use the same Wordnet tree structure from which the semantic distance matrix  $S$  is derived. At each edge in the tree, we train a linear SVM in the manner described in [27]. Note that both our semantic sharing method and that of Marszalek and Schmid are provided with the same semantic information. Hence, by comparing the two approaches we can see which makes more efficient use of the semantic information.

These three approaches are evaluated on the *CIFAR* and *High-res* datasets in Figures 9 and 10 respectively. The latter dataset also shows the semi-supervised scheme without sharing. The two figures show consistent results that clearly demonstrate: (i) the addition of semantic information helps – both the H-SVM and SSL with sharing beat the methods without it; (ii) our sharing framework is superior to that of Marszalek and Schmid [27].

## 5 Summary and Future Work

We have introduced a very simple mechanism for sharing training labels between classes. Our experiments on a variety of datasets demonstrate that it gives significant benefits in situations where there are many classes, a common occurrence in large image collections. We have shown how semantic sharing can be combined with simple classifiers to operate on large datasets up to 75,000 classes and 79 million images. Furthermore, our experiments clearly demonstrate that our sharing approach outperforms other methods that use semantic information when constructing the classifier. While the semantic sharing matrix from Wordnet has proven effective, a goal of future work would be to learn it directly from the data.

## References

1. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large database for non-parametric object and scene recognition. *IEEE PAMI* 30, 1958–1970 (2008)
2. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. *IJCV* 77, 157–173 (2008)

3. van Ahn, L.: The ESP game (2006)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR'09 (2009)
5. Fellbaum, C.: Wordnet: An Electronic Lexical Database. Bradford Books (1998)
6. Biederman, I.: Recognition-by-components: A theory of human image understanding. *Psychological Review* 94, 115–147 (1987)
7. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
8. Li, L.J., Wang, G., Fei-Fei, L.: Imagenet. In: CVPR (2007)
9. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV, vol. 2, pp. 1816–1823 (2005)
10. Berg, T., Forsyth, D.: Animals on the web. In: CVPR, pp. 1463–1470 (2006)
11. Caruana, R.: Multitask learning. *Machine Learning* 28, 41–75 (1997)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324 (1998)
13. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via ECOCs. *JAIR* 2, 263–286 (1995)
14. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: Proc. of the 2004 IEEE CVPR (2004)
15. Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: CVPR, vol. (1), pp. 3–10 (2006)
16. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (to appear, 2004)
17. Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Learning hierarchical models of scenes, objects, and parts. In: Proceedings of the IEEE International Conference on Computer Vision, Beijing (to appear, 2005)
18. Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. *Neural Computation* 12, 1247–1283 (2000)
19. Bart, E., Ullman, S.: Cross-generalization: learning novel classes from a single example by feature replacement. In: CVPR (2005)
20. Miller, E., Matsakis, N., Viola, P.: Learning from one example through shared densities on transforms. In: CVPR, vol. 1, pp. 464–471 (2000)
21. Quattoni, A., Collins, M., Darrell, T.: Transfer learning for image classification with sparse prototype representations. In: CVPR (2008)
22. Budanitsky, Hirst: Evaluating wordnet-based measures of lexical semantic relatedness. In: *Computational Linguistics* (2006)
23. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-Supervised Learning*. MIT Press, Cambridge (2006)
24. Schoelkopf, B., Smola, A.: *Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge (2002)
25. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)
26. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: NIPS (2009)
27. Marszalek, M., Schmid, C.: Semantic hierarchies for visual object recognition. In: CVPR (2007)

# Efficient Object Category Recognition Using Classemes

Lorenzo Torresani<sup>1</sup>, Martin Szummer<sup>2</sup>, and Andrew Fitzgibbon<sup>2</sup>

<sup>1</sup> Dartmouth College, Hanover, NH, USA

[www.cs.dartmouth.edu/~lorenzo](http://www.cs.dartmouth.edu/~lorenzo)

<sup>2</sup> Microsoft Research, Cambridge, United Kingdom

<http://www.research.microsoft.com/~{szummer,awf}>

**Abstract.** We introduce a new descriptor for images which allows the construction of efficient and compact classifiers with good accuracy on object category recognition. The descriptor is the output of a large number of weakly trained object category classifiers on the image. The trained categories are selected from an ontology of visual concepts, but the intention is not to encode an explicit decomposition of the scene. Rather, we accept that existing object category classifiers often encode not the category *per se* but ancillary image characteristics; and that these ancillary characteristics can combine to represent visual classes unrelated to the constituent categories' semantic meanings.

The advantage of this descriptor is that it allows object-category queries to be made against image databases using efficient classifiers (efficient at test time) such as linear support vector machines, and allows these queries to be for novel categories. Even when the representation is reduced to 200 bytes per image, classification accuracy on object category recognition is comparable with the state of the art (36% versus 42%), but at orders of magnitude lower computational cost.

## 1 Introduction

The accuracy of object category recognition is improving rapidly, particularly if the goal is to retrieve or label images where the category of interest is the primary subject of the image. However, existing techniques do not scale well to searching in large image collections. This paper identifies three requirements for such scaling, and proposes a new descriptor which satisfies them.

We suggest that interesting large-scale applications must recognize **novel categories**. This means that a new category can be presented as a set of training images, and a classifier learned from these new images can be run efficiently against the large database. Note that kernel-based classifiers, which represent the current state of the art, do not satisfy this requirement because the (kernelized) distance between each database image and (a subset of) the novel training images must be computed. Without the novel-category requirement, the problem is trivial—the search results can be precomputed by running the known category detector on each database image at ingestion time, and storing the results as inverted files.

**Table 1. Highly weighted classemes.** Five classemes with the highest LP- $\beta$  weights for the retrieval experiment, for a selection of Caltech 256 categories. Some may appear to make semantic sense, but it should be emphasized that our goal is simply to create a useful feature vector, not to assign semantic labels. The somewhat peculiar classeme labels reflect the ontology used as a source of base categories.

New category	Highly weighted classemes				
<b>cowboy-hat</b>	helmet	sports_track	cake_pan	collectible	muffin_pan
<b>duck</b>	bomber_plane	body_of_water	swimmer	walking	straight
<b>elk</b>	figure_skater	bull_male_herd_animal	cattle	gravesite	dead_body
<b>frisbee</b>	watercraft_surface	scsi_cable	alarm_clock	hindu	servicing_tray
<b>trilobite-101</b>	convex_thing	mined_area	cdplayer	roasting_pan	western_hemisphere_person
<b>wheelbarrow</b>	taking_care_of_something	baggage_porter	canopy_closure_open	rowing_shell	container_pressure_barrier

Large-scale recognition benefits from a **compact descriptor** for each image, for example allowing databases to be stored in memory rather than on disk. The descriptor we propose is 2 orders of magnitude more compact than the state of the art, at the cost of a small drop in accuracy. In particular, performance of the state of the art with 15 training examples is comparable to our most compact descriptor with 30 training examples.

The ideal descriptor also provides good results with **simple classifiers**, such as linear SVMs, decision trees, or tf-idf, as these can be implemented to run efficiently on large databases.

Although a number of systems satisfy these desiderata for object instance or place recognition [18,9] or for whole scene recognition [26], we argue that no existing system has addressed these requirements in the context of object category recognition.

The system we propose is a form of classifier combination, the components of the proposed descriptor are the outputs of a set of predefined category-specific classifiers applied to the image. The obvious (but only partially correct) intuition is that a novel category, say **duck**, will be expressed in terms of the outputs of base classifiers (which we call “classemes”), describing either objects similar to ducks, or objects seen in conjunction with ducks. Because these base classifier outputs provide a rich coding of the image, simple classifiers such as linear SVMs can approach state-of-the-art accuracy, satisfying the requirements listed above. However, the reason this descriptor will work is slightly more subtle. It is not required or expected that these base categories will provide useful semantic labels, of the form **water**, **sky**, **grass**, **beak**. On the contrary, we work on the assumption that modern category recognizers are essentially quite dumb; so a **swimmer** recognizer looks mainly for water texture, and the **bomber\_plane** recognizer contains some tuning for “C” shapes corresponding to the airplane nose, and perhaps the “V” shapes at the wing and tail. Even if these recognizers are perhaps overspecialized for recognition of their nominal category, they can still provide useful building blocks to the learning algorithm that learns to recognize

the novel class `duck`. Table 1 lists some highly-weighted classemes used to describe an arbitrarily selected subset of the Caltech256 categories. Each row of the table may be viewed as expressing the category as a weighted sum of building blocks; however the true building blocks are not the classeme labels that we can see, but their underlying dumb components, which we cannot. To complete the duck example, it is a combination of `body_of_water`, `bomber_plane`, `swimmer`, as well as `walking` and `straight`. To gain an intuition as to what these categories actually represent, Figure 2 shows the training sets for the latter two. Examining the training images, we suggest that `walking` may represent “inverted V outdoors” and `straight` might correspond to “clutter and faces”.

## 2 Background

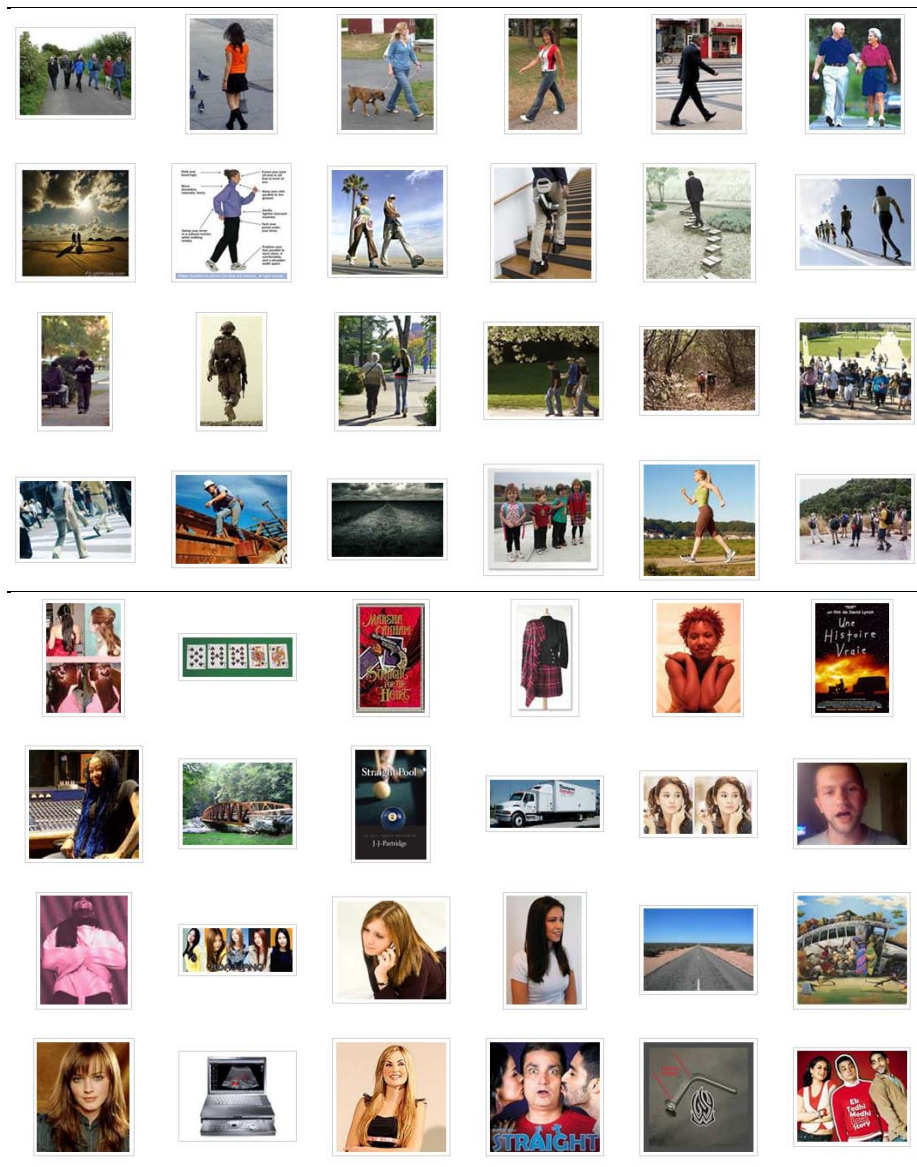
Before describing the details of the system, and experimental investigations, we shall briefly summarize related literature, and discuss how existing systems fit the requirements.

The closest existing approach is probably image representation via *attributes* [5,11]. Here object categories are described by a set of boolean attributes, such as “has beak”, “no tail”, “near water”. Classifiers for these attributes are built by acquiring labels using Amazon’s Mechanical Turk. In contrast, we do not design our attributes to have specific semantic meanings, but expect meaning to emerge from intersections of properties, and we obtain training data directly from web image search without human cleanup. Furthermore, prior attribute-based methods have relied on a “zero-shot” learning approach: instead of *learning* a classifier for a novel category from training examples, a user designs the classifier by listing attributes, limiting such systems to categories for which humans can easily extract attributes, and increasing the workload on the user even for such categories.

Our approach is also evocative of Malisiewicz and Efors’s “Recognition by Association” [14], in which object classes are represented by sets of object *instances* to which they are associated. In contrast, we represent object classes as a combination of other object *classes* to which they are related. This change of viewpoint allows us to use the powerful classifiers produced by recent advances in object category recognition.

Because we represent images by a (relatively) low-dimensional feature vector, our approach is related to dimensionality reduction techniques exemplified by *semantic hashing* [20,26]. These data-driven techniques find low-dimensional, typically nonlinear, projections of a large feature vector representing each image, such that the low-dimensional vectors are an effective proxy for the original. These techniques can achieve tremendous compressions of the image (for example to 64 bits [26]), but are of course correspondingly lossy, and have not been shown to be able to retain category-level information.

It is also useful to make a comparison to systems which, while less related in form, represent the state of the art in object category recognition. The assessment is thus in terms of how far the existing systems meet the requirements we



**Fig. 1. Classeme training images.** A subset of the training images for two of the 2659 classemes: *walking*, and *straight*. The top 150 training images are downloaded from Bing image search with no filtering or reranking. As discussed in the text, we do not require classemes categories to have a semantic relationship with the novel class; but to contain some building blocks useful for classification.



have set out. In the discussion below, let  $N$  be the size of the test set (i.e. the image database, which may in principle be very large). Let  $n$  be the number of images in the training set, typically in the range 5 – 100 per class. Let  $d$  be the dimensionality of the representation stored for each image. For example, if a histogram of visual words is stored,  $d$  is the minimum of the number of words detected per image and the vocabulary size. For a GIST descriptor [19],  $d$  is of the order of 1000. For multiple-kernel techniques [6],  $d$  might be of the order of 20,000. For the system in this paper,  $d$  can be as low as 1500, while still leveraging all the descriptors used in the multiple-kernel technique. Note that although we shall later be specific about the number of bits per element of  $d$ , this is not required for the current discussion.

Boiman *et al.* [2] shows one of the most intriguing results on the Caltech 256 benchmark: a nearest-neighbour-like classifier on low-level feature descriptors produces excellent performance, especially with small training sets. Its training cost is effectively zero: assemble a bag of descriptors from the supplied training images (although one might consider building a kd-tree or other spatial data structure to represent large training sets). However, the test-time algorithm requires that each descriptor in the *test* image be compared to the bag of descriptors representing the class, which has complexity  $O(nd)$ . It may be possible to build a kd-tree for the test set, and reverse the nearest-neighbor lookups, but the metric is quite asymmetric, so it is not at all clear that this will preserve the properties of the method.

Gehler and Nowozin [6] represents the state of the art in classification accuracy on the Caltech 256 benchmarks, using a kernel combination classifier. However, training this classifier is expensive, and more importantly, test-time evaluation requires that several kernels be evaluated between each test image and several images of the training set. Note that these kernels cannot be precomputed, because in our problem the training set is different for every new query. Therefore the complexity is again  $O(nd)$ , but with large  $d$ , and a relatively large constant compared to the nearest-neighbor approach.

Another class of related techniques is the use of classifier combination other than multiple-kernel approaches. Zehnder *et al.* [27] build a classifier cascade which encourages feature sharing, but again requires the set of classes to be predefined, as is true for Griffin and Perona [7] and Torralba *et al.* [23]. Heitz *et al.* [8] propose to learn a general cascade similar to ours (although with a different goal), but our approach simplifies training by pre-training the first layer, and simplifies test by successfully working with simple top-layer classifiers.

### 3 Method Overview

Our approach is now described precisely, but briefly, with more details supplied in §4. There are two distinct stages: once-only classem learning; followed by any number of object-category-related learning tasks. Note that there are distinct training sets in each of the two stages.

### 3.1 Classeseme Learning

A set of  $C$  category labels is drawn from an appropriate term list. For each category  $c \in \{1..C\}$ , a set of training images is gathered by issuing a query on the category label to an image search engine.

A one-versus-all classifier  $\phi_c$  is trained for each category. The classifier output is real-valued, and is such that  $\phi_c(\mathbf{x}) > \phi_c(\mathbf{y})$  implies that  $\mathbf{x}$  is more similar to class  $c$  than  $\mathbf{y}$  is. Given an image  $\mathbf{x}$ , then, the feature vector (descriptor) used to represent  $\mathbf{x}$  is the *classeseme vector*  $\mathbf{f}(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_C(\mathbf{x})]$ .

Given the classeseme vectors for all training images, it may be desired to perform some feature selection on the descriptors. We shall assume this has been done in the sequel, and simply write the classeseme vector in terms of a reduced dimensionality  $d \leq C$ , so  $\mathbf{f}(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_d(\mathbf{x})]$ . Where  $d$  is not specified it may be assumed that  $d = C$ .

Given the parameters of the  $\phi_c$ , the training examples used to create the classesemes may be discarded. We denote by  $\Phi$  the set of functions  $\{\phi_c\}_{c=1}^d$ , which encapsulates the output of the classeseme learning, and properly we shall write  $\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}; \Phi)$ .

### 3.2 Using the Classesemes

Given  $\Phi$ , the rest of our approach is conventional. A typical situation might be that a new object category, or set of categories, is defined by a set of training images (note again that this is a new set of training images, unrelated to those used to build  $\Phi$ ). The training images are converted to classeseme vectors, and then any classifier can be trained taking the classeseme vectors as input. As we show in experiments, the features are sufficiently powerful that simple and fast classifiers applied to the classesemes can give accuracies commensurate with much more expensive classifiers applied to the low-level image features. Useful candidate classifiers might be those which make a sparse linear combination of input features, so that test cost is a small fraction of  $d$  per image; or predicate-based classifiers so that test images with nonzero score can be retrieved rapidly using inverted files [18,24], achieving test complexity sublinear in  $N$ , the size of the test set.

## 4 Further Details

Several details are now expanded.

### 4.1 Selecting Category Labels

The set of category labels used to build the classesemes should consist primarily of visual concepts. This will include concrete nouns, but may also include more abstract concepts such as “person working”. Although we know of no general rules for category selection, the category labels should probably be chosen to be representative of the type of applications in which one plans to use the

descriptors. As we considered general-category image search as a target application domain, we selected concepts from the Large Scale Concept Ontology for Multimedia (LSCOM) [17]. The LSCOM categories were developed specifically for multimedia annotation and retrieval, and have been used in the TRECVID video retrieval series. This ontology includes concepts selected to be useful, observable and feasible for automatic detection, and as such are likely to form a good basis for image retrieval and object recognition tasks. We took the LSCOM CYC ontology dated 2006-06-30 [13], which contains 2832 unique categories. We removed 97 categories denoting abstract groups of other categories (marked in angle brackets in [13]), and then removed plural categories that also occurred as singulars, and some people-related categories which were effectively near-duplicates, and arrived at  $C = 2659$  categories. Some examples have already been seen in table 1. We were conservative in removing categories because, as discussed in the introduction, it is not easy to predict *a priori* what categories will be useful.

## 4.2 Gathering Category Training Data

For each category label, a set of training images was gathered by taking the top 150 images from the [bing.com](http://bing.com) image search engine. For a general application these examples would not need to be manually filtered in any way, but in order to perform fair comparisons against the Caltech image database, near duplicates of images in that database were removed by a human-supervised process. Conversely, we did not remove overlap between the *classeme terms* and the Caltech categories (28 categories overlap, see supplementary data on [25]), as a general-purpose system can expect to see overlap on a small number of queries. However, we do include one test (CsvmN, figure 2) which shows no significant drop in performance by removing these terms.

## 4.3 Learning Classifiers $\phi_c$

The learning algorithm used for the  $\phi(\cdot)$  is the LP- $\beta$  kernel combiner of Gehler and Nowozin [6]. They used 39 kernels, but we reduced this to 13 for experimentation. We employed kernels defined in terms of  $\chi^2$  distance between feature vectors, i.e.  $k(x, x') = \exp(-\chi^2(x, x')/\gamma)$ , using the following 13 feature types:

- *Kernel 1: Color GIST*,  $d_1 = 960$ . The GIST descriptor [19] is applied to color images. The images were resized to  $32 \times 32$  (aspect ratio is not maintained), and then orientation histograms were computed on a  $4 \times 4$  grid. Three scales were used with the number of orientations per scale being 8, 8, 4.
- *Kernels 2-5: Pyramid of Histograms of Oriented Gradients*,  $d_{2..5} = 1700$ . The PHOG descriptor [4] is computed using 20 bins at four spatial pyramid scales.
- *Kernels 6-9: PHOG ( $2\pi$  unwrapped)*,  $d_{6..9} = 3400$ . These features are obtained by using unoriented gradients quantized into 40 bins at four spatial pyramid scales.

- *Kernels 10-12: Pyramid self-similarity*,  $d_{10..12} = 6300$ . The Shechtman and Irani self-similarity descriptor [21] was computed as described by Bosch [3]. This gives a 30-dimensional descriptor at every 5th pixel. We quantized these descriptors into 300 clusters using  $k$ -means, and a pyramid histogram was recorded with three spatial pyramid levels.
- *Kernel 13: Bag of words*.  $d_{13} = 5000$ . SIFT descriptors [12] were computed at interest points detected with the Hessian-Affine detector [16]. These descriptors were then quantized using a vocabulary of size 5000, and accumulated in a sparse histogram.

A binary LP- $\beta$  classifier was trained for each classeme, using a setup following the one described in section 7 of [6] in terms of kernel functions, kernel parameters, values of  $\nu$  and number of cross validations. The only difference is that the objective of their equation (4) was modified in order to handle the uneven training set sizes. We used  $N_+ = 150$  images as positive examples, and one image chosen at random from each of the other training sets as negative examples, so  $N_- = C - 1$ . The objective was modified by scaling the positive entries in the cost vector by  $(\nu N_+)$  and the negative entries by  $(\nu N_-)$ . The cross-validation yields a per-class validation score which is used for feature selection.

#### 4.4 Feature Selection

We perform some simple dimensionality reduction of the classeme vectors  $\mathbf{f}$  as follows. The classemes are sorted in increasing order of cross-validation error. Given a desired feature dimensionality,  $d$ , the reduced classeme vector is then simply the first  $d$  components  $\mathbf{f}(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_d(\mathbf{x})]$ . Again in situations where  $d$  is not specified it may be assumed that  $d = C$

#### 4.5 Classeme Quantization

For a practical system, the classeme vectors should not be stored in double precision, but instead an explicit quantization of the values should be used. This may be achieved by a simple quantization, or by defining binary “decision stumps” or predicates. Quantization can be performed either at novel-category learning time (i.e. on the novel training set) or at classeme-learning time. For 1-bit quantization we just thresholded at 0. For higher numbers, we use the following “histogram-equalized” quantization. Given a training set of classeme vectors  $\{\mathbf{f}_i\}_{i=1}^n$ , write  $\mathbf{f}_i = [\phi_{ik}]_{k=1}^d$ . Write the rows of the matrix  $[\mathbf{f}_1, \dots, \mathbf{f}_n]$  as  $\mathbf{r}_k = [\phi_{ik}]_{i=1}^n$ . To quantize to  $Q$  levels, quantization centres  $z_{iq}$  are chosen as follows:  $\mathbf{r}'_k = \text{sort}(\mathbf{r}_k)$ , defining a row-sorted matrix  $\phi'_{ik}$ . Then make the set  $Z_k = \{\phi'_{[nq/(Q+1)],k}\}_{q=1}^Q$ , and each value  $\phi_{ik}$  is replaced by the closest value in  $Z_k$ .

## 5 Experiments

Given the simplicity of the approach, the first question that naturally arises is how it compares to the state-of-the-art recognition approaches. Here we compare

to the LP- $\beta$  kernel combiner as this is the current front-runner. Note that the key metric here is performance drop with respect to LP- $\beta$  with 13 kernels, as this means that the base features are the same between LP- $\beta$  and classemes.

As our classifiers introduce an extra step in the recognition pipeline, performance might be expected to suffer from a “triangle inequality”: the raw kernel combiner can optimize kernels on the  $d_{LP}$  features directly, while the classemes classifiers are forced to go via the  $d$  classemes. We will show that this does happen, but to a small enough extent that the classemes remain competitive with the state of the art, and are much better than the closest “efficient” system.

There are two main experiments. In the first, we wish to assess the representational power of classemes with respect to existing methods, so we use the standard Caltech 256 accuracy measure, with multiclass classifiers trained on all classes. In the second, we want to test classemes in a framework closer to their intended use, so we train one-vs-all classifiers on each Caltech class, and then report precision on ranking a set of images including distractors from the other classes.

### 5.1 Experiment 1: Multiclass Classification

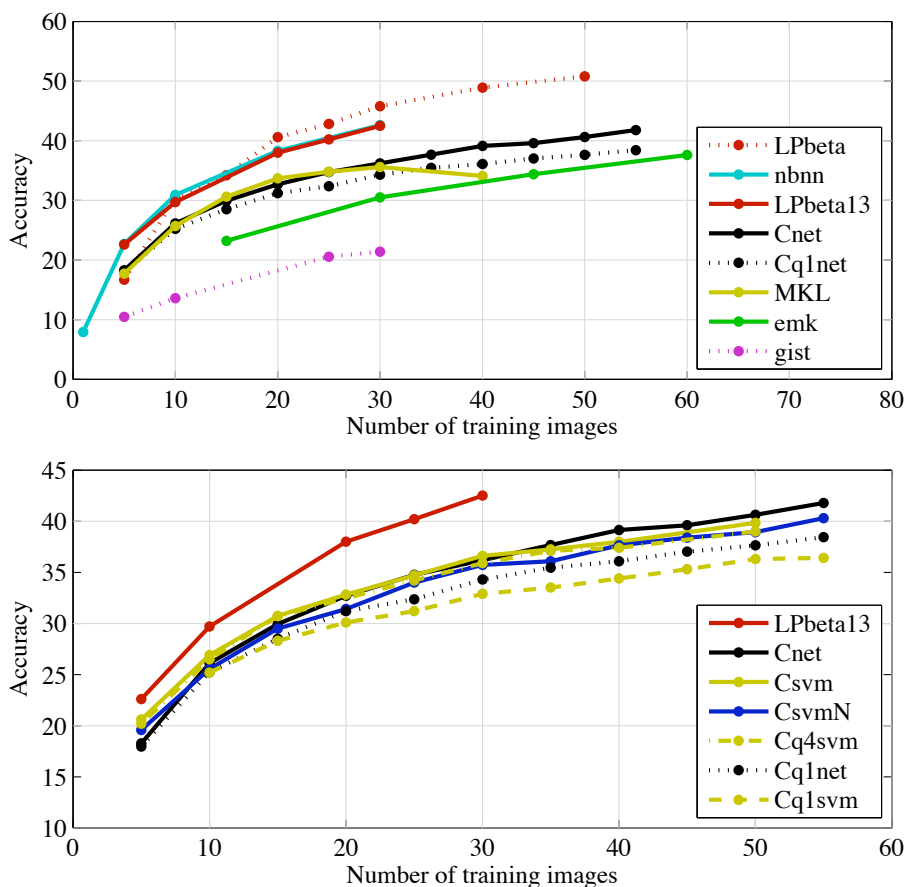
To use classemes for multiclass classification, several strategies were implemented: multiclass SVMs [10], neural networks, decision forests [22] and a nearest-neighbour classifier. Comments on each of these follow. The variable  $T$  is the number of training examples per class. Figures 2 and 3 summarize the results.

Multiclass SVMs were trained using the SVMlight software [10], with regularization parameter  $\lambda = 3000$ , and  $d = 1500$ . The regularization parameter was determined by cross-validation for  $T = 15$ , and fixed for all subsequent tests.

Decision trees were trained using a standard information gain criterion. Because the small training set size limits the tree depth (depth 9 was found by cross-validation at  $T = 15$ ), decision forests were found to require large forests (around 50 trees), and were not included in subsequent testing. Similarly, a nearest-neighbour classifier was tested and found to give low performance, so it is not included in these results, but complete results can be seen at [25].

A standard 1-hidden-layer network with tanh nonlinearities and a softmax over 256 outputs was trained. The number of hidden units ranged between 500 and 1000, chosen on a validation set, and training used an  $L_1$  weight decay regularizer fixed at 0.1 and was stopped early at 40–80 iterations.

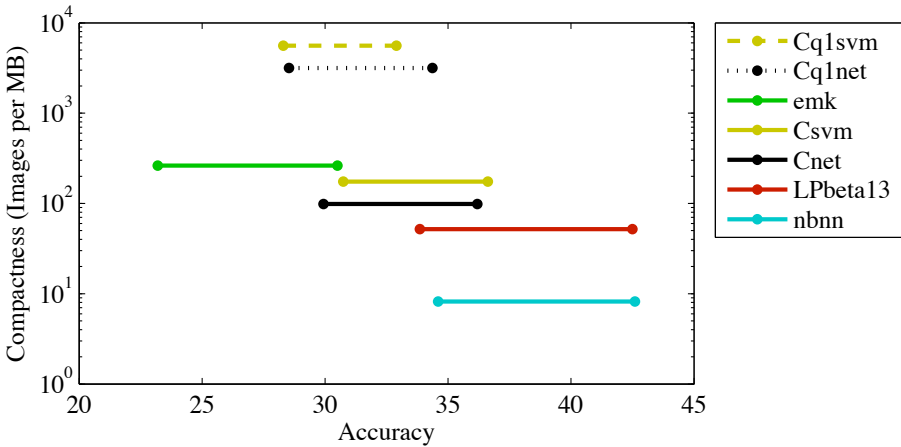
For each class, the number of training images was varied, and 25 test images were used. Performance is quoted as the mean of average accuracy per class as in [6], and plotted in figure 2. It can be seen that the classemes-based neural network (Cnet) and SVM classifiers (Csvm) beat all but LP- $\beta$  and NBNN. However the size of the representation is considerably reduced for classemes compared to LP- $\beta$  and NBNN: 2.5K for classemes versus 23K and 128K respectively. Furthermore, the training and test times of our approach are considerably lower than LP- $\beta$ : training the multiclass classifier Csvm with 5 examples for each Caltech class takes about 9 minutes on a AMD Opteron Processor 280 2.4GHz while



**Fig. 2. Caltech 256.** A number of classifiers are compared on the Caltech 256 dataset. The key claim is this: on 30 training examples, and using the same underlying features, Cnet1q1 and Csvm have 36% accuracy, and LPbeta13 has 42% accuracy, but the classeme-based systems are orders of magnitude faster to train and test.

(**Top**): Classeme neural net compared to results in the literature: **LPbeta** [6], **NBNN**: Naive Bayes Nearest Neighbor [2]; **MKL**: Multiple Kernel learning, as implemented in [6]; **EMK**: Efficient Match Kernel [1]. In addition we add our implementations of: **LPbeta13** ( $LP-\beta$  on our base features §4.3); **GIST**: One-vs-all SVM on GIST feature alone; **Cnet**: Neural network trained on floating point classeme vector; **Cq1net**: Neural network on 1 bit-per-channel (1bpc) classeme vector.

(**Bottom**): Comparison of classeme-based classifiers. (Except LPbeta13, included for reference). **Csvm**: SVM, floating point,  $d = 1500$ ; **CsvmN**: SVM, floating point, Caltech terms removed from training (§4.2); **Cq4svm**: SVM, input quantized to 4 bits per channel (bpc),  $d = 1500$ ; **Cq1svm**: SVM, input quantized to 1 bit,  $d = 1500$ .



**Fig. 3.** Accuracy versus compactness of representation on Caltech-256. On both axes, higher is better. (Note logarithmic  $y$ -axis). The lines link performance at 15 and 30 training examples.

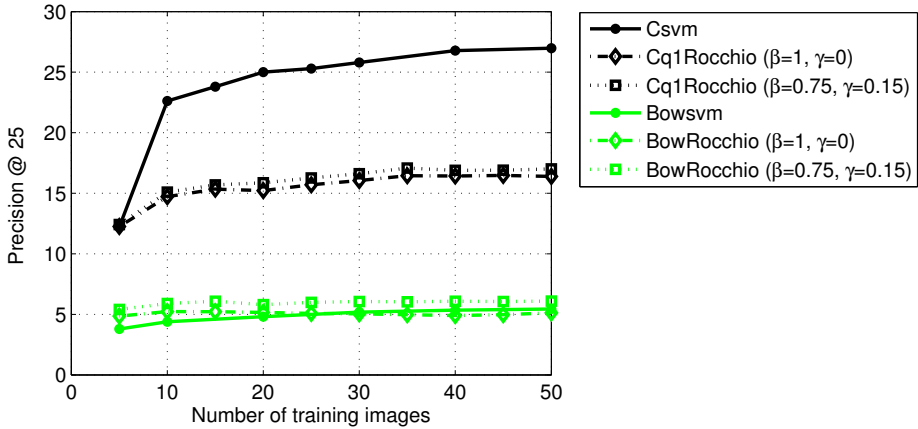
the method of [6] requires more than 23 hours on the same machine; predicting the class of a test example takes 0.18ms with our model and 37ms with LP- $\beta$ .

Furthermore, when moving from floating point classes (Csvm) to a quantization of 4 bits per channel (Cq4svm) the change in accuracy is negligible. Accuracy drops by 1–2 percentage points using a 1 bit per channel neural network (Cq1net, 312 bytes per image), and 1–2 more using a 1bpc SVM (Cq1svm,  $d = 1500$ , 187.5 bytes per image). This storage requirement increases the number of images that can be stored in an index by a factor of 100 over LP- $\beta$ , which is especially significant for RAM-based indices.

Also plotted for reference is the accuracy of GIST as a single feature, being an important contributor to LP- $\beta$ 's kernel pool. We note that the GIST vector at 960 bytes is already much longer than the 187.5 bytes of Cq1svm while being much less accurate.

## 5.2 Experiment 2: Retrieval

The retrieval experiment attempts to gain insight into the behaviour of classes in a retrieval task. The test database is a concatenation of 25 images from each Caltech category. A query against the database is specified by a set of training images taken from one category, and the retrieval task is to order the database by similarity to the query. Success is measured as precision at 25: the proportion of the top 25 images which are in the same category as the query set. The maximum score is 1, obtained if all the matching images are ranked above all the distractors. For this experiment, we compare classes with bags of visual words (BOW), which are a popular model for efficient image retrieval. We use as BOW features the quantized SIFT descriptors of Kernel 13.



**Fig. 4. Retrieval.** Percentage of the top 25 in a 6400-document set which match the query class. Random performance is 0.4%.

We consider two different retrieval methods. The first method is a linear SVM learned for each of the Caltech classes using the one-vs-all strategy. We compare these classifiers to the Rocchio algorithm [15], which is a classic information retrieval technique for implementing relevance feedback. In order to use this method we represent each image as a document vector  $\mathbf{d}(\mathbf{x})$ . In the case of the BOW model,  $\mathbf{d}(\mathbf{x})$  is the traditional *tf-idf*-weighted histogram of words. In the case of classemes instead, we define  $\mathbf{d}(\mathbf{x})_i = [\phi_i(\mathbf{x}) > 0] \cdot \text{idf}_i$ , i.e.  $\mathbf{d}(\mathbf{x})$  is computed by multiplying the binarized classemes by their inverted document frequencies. Given, a set of relevant training images  $D_r$ , and a set of non-relevant examples  $D_{nr}$ , Rocchio’s algorithm computes the document query

$$\mathbf{q} = \beta \frac{1}{|D_r|} \sum_{\mathbf{x}_r \in D_r} \mathbf{d}(\mathbf{x}_r) - \gamma \frac{1}{|D_{nr}|} \sum_{\mathbf{x}_{nr} \in D_{nr}} \mathbf{d}(\mathbf{x}_{nr}) \quad (1)$$

where  $\beta$  and  $\gamma$  are scalar values. The algorithm then retrieves the database documents having highest *cosine similarity* with this query. In our experiment, we set  $D_r$  to be the training examples of the class to retrieve, and  $D_{nr}$  to be the remaining training images. We report results for two different settings:  $(\beta, \gamma) = (0.75, 0.15)$ , and  $(\beta, \gamma) = (1, 0)$  corresponding to the case where only positive feedback is used.

Figure 4 shows that methods using classemes consistently outperform the algorithms based on traditional BOW features. Furthermore, SVM yields much better precision than Rocchio’s algorithm when using classemes. Note that these linear classifiers can be evaluated very efficiently even on large data sets; furthermore, they can also be trained efficiently and thus used in applications requiring fast query-time learning: for example, the average time required to learn a one-vs-all SVM using classemes is 674 ms when using 5 training examples from each Caltech class.



## 6 Discussion

We have introduced a new descriptor for images which is intended to be useful for high-level object recognition. By using the noisy training data from web image search in a novel way: to train “category-like” classifiers, the descriptor is essentially given access to knowledge about what humans consider “similar” when they search for images on the web (note that most search engines are considered to use “click-log” data to rank their image search results, so the results do reflect human preferences). We have shown that this knowledge is effectively encoded in the classeme vector, and that this vector, even when quantized to below 200 bytes per image, gives competitive object category recognition performance.

An important question is whether the weakly trained classemes actually do contain any semantic information. We have emphasized throughout the paper that this is not the main motivation for their use, and we do so again here. It may be that one might view the classemes as a form of highly nonlinear random projection, and it is interesting future work to see if something close to random splits will yield equivalent performance.

We have focused here on object category recognition as characterized by the Caltech 256 training data, which we consider adequate for clip-art search, but which will not be useful for, for example, home photo retrieval, or object indexing of surveillance footage. It should be straightforward to retrain the classemes on images such as the PASCAL VOC images, but a sliding-window approach would probably be required in order to achieve good performance.

Several avenues remain open to improve these results. Our feature selection from among the 2659 raw features is currently very rudimentary, and it may be helpful to apply a sparse classifier. The various hashing techniques can immediately be applied to our descriptor, and might result in considerable reductions in storage and computational cost.

Additional material including the list of classemes, the retrieved training images, and precomputed classeme vectors for standard datasets, may be obtained from [25].

## References

1. Bo, L., Sminchisescu, C.: Efficient Match Kernel between Sets of Features for Visual Recognition. In: *Adv. in Neural Inform. Proc. Systems* (December 2009)
2. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: *Proc. Comp. Vision Pattern Recogn., CVPR* (2008)
3. Bosch, A.: Image classification using rois and multiple kernel learning (2010), [http://eia.udg.es/~aboschr/Publicacions/bosch08a\\_preliminary.pdf](http://eia.udg.es/~aboschr/Publicacions/bosch08a_preliminary.pdf)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, vol. 1, pp. 886–893 (2005)
5. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: *Proc. Comp. Vision Pattern Recogn. (CVPR)*, pp. 1778–1785 (2009)
6. Gehler, P.V., Nowozin, S.: On feature combination for multiclass object classification. In: *Intl. Conf. Computer Vision* (2009)

7. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: Proc. Comp. Vision Pattern Recogn., CVPR (2008)
8. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded classification models: Combining models for holistic scene understanding. In: Adv. in Neural Inform. Proc. Systems (NIPS), pp. 641–648 (2008)
9. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European Conf. Comp. Vision (October 2008)
10. Joachims, T.: An implementation of support vector machines (svms) in c (2002)
11. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Proc. Comp. Vision Pattern Recogn., CVPR (2009)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. Intl. Jnl. of Computer Vision 60(2), 91–110 (2004)
13. LSCOM: Cyc ontology dated (2006-06-30),  
<http://lastlaugh.inf.cs.cmu.edu/lscom/ontology/LSCOM-20060630.txt>,  
<http://www.lscom.org/ontology/index.html>
14. Malisiewicz, T., Efros, A.A.: Recognition by association via learning per-exemplar distances. In: Proc. Comp. Vision Pattern Recogn., CVPR (2008)
15. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
16. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Intl. Jnl. of Computer Vision 60(1), 63–86 (2004)
17. Naphade, M., Smith, J.R., Tesci, J., Chang, S.F., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J.: Large-scale concept ontology for multimedia. IEEE MultiMedia 13(3), 86–91 (2006)
18. Nistér, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. Comp. Vision Pattern Recogn. (CVPR), pp. 2161–2168 (2006)
19. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Visual Perception, Progress in Brain Research 155 (2006)
20. Salakhutdinov, R., Hinton, G.: Semantic hashing. In: SIGIR Workshop on Information Retrieval and Applications of Graphical Models (2007)
21. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: Proc. Comp. Vision Pattern Recogn. CVPR (June 2007)
22. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Proc. Comp. Vision Pattern Recogn., CVPR (2008)
23. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5), 854–869 (2007)
24. Torresani, L., Szummer, M., Fitzgibbon, A.: Learning query-dependent prefilters for scalable image retrieval. In: Proc. Comp. Vision Pattern Recogn. (CVPR), pp. 2615–2622 (2009)
25. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classesemes (2010),  
<http://www.cs.dartmouth.edu/~lorenzo/projects/classesemes>
26. Weiss, Y., Torralba, A.B., Fergus, R.: Spectral hashing. In: Adv. in Neural Inform. Proc. Systems (NIPS), pp. 1753–1760 (2008)
27. Zehnder, P., Koller-Meier, E., Gool, L.V.: An efficient shared multi-class detection cascade. In: British Machine Vision Conf. (2008)

# Practical Autocalibration

Riccardo Gherardi and Andrea Fusiello

Dipartimento di Informatica, Università di Verona  
Strada Le Grazie 15, 37134 Verona, Italy  
`name.surname@univr.it`

**Abstract.** As it has been noted several times in literature, the difficult part of autocalibration efforts resides in the structural non-linearity of the search for the plane at infinity. In this paper we present a robust and versatile autocalibration method based on the enumeration of the inherently bounded space of the intrinsic parameters of two cameras in order to find the collineation of space that upgrades a given projective reconstruction to Euclidean. Each sample of the search space (which reduces to a finite subset of  $\mathbb{R}^2$  under mild assumptions) defines a consistent plane at infinity. This in turn produces a tentative, approximate Euclidean upgrade of the whole reconstruction which is then scored according to the expected intrinsic parameters of a Euclidean camera. This approach has been compared with several other algorithms on both synthetic and concrete cases, obtaining favourable results.

**Keywords:** Autocalibration, Self-calibration.

## 1 Introduction

Autocalibration (a.k.a. self-calibration) has generated a lot of theoretical interest since its introduction in the seminal paper by Maybank and Faugeras [1]. The attention spawned by the problem however is inherently practical, since it eliminates the need for off-line calibration and enables the use of content acquired in an uncontrolled environment. Modern computer vision has partly sidestepped the issue using ancillary information, such as EXIF tags embedded in some image formats. Such data unfortunately is not always guaranteed to be present or consistent with its medium, and does not extinguish the need for reliable autocalibration procedures.

Lots of published methods rely on equations involving the dual image of the absolute quadric (DIAQ), introduced by Triggs in [2]. Earliest approaches for variable focal lengths were based on linear, weighted systems [3,4], solved directly or iteratively [5]. Their reliability were improved by more recent algorithms, such as [6], solving super-linear systems while forcing directly the positive definiteness of the DIAQ. Such enhancements were necessary because of the structural non-linearity of the task: for this reason the problem has also been approached using branch and bound schemes, based either on the Kruppa equations [7], dual linear autocalibration [8] or the modulus constraint [9].

The algorithm described in [10] shares with the branch and bound approaches the guarantee of convergence; the non-linear part, corresponding to the localization of the plane at infinity, is solved exhaustively after having used the cheiral inequalities to compute explicit bounds on its location.

The technique we are about to describe is closely related to the latter: first, we derive the location of the plane at infinity given two perspective projection matrices and a guess on their intrinsic parameters, and subsequently use this procedure to iterate through the space of camera intrinsic parameters looking for the best collineation that makes the reconstruction Euclidean. The search space is inherently bounded by the finiteness of the acquisition devices; each sample and the corresponding plane at infinity define a collineation of space whose likelihood can be computed evaluating skew, aspect ratio, principal point and related constraints for each transformed camera. The best solution is eventually refined via non-linear least squares.

Such approach has several advantages: it's fast, easy to implement and reliable, since a reasonable solution can always be found in non-degenerate configurations, even in extreme cases such as when autocalibrating just two cameras.

## 2 Method

As customary, we assume being given a projective reconstruction  $\{P_i; X_j\}$   $i = 1 \dots n$ ;  $j = 1 \dots m$ . The purpose of autocalibration is therefore to find the collineation of space  $H$  such that  $\{P_i H; H^{-1} X_j\}$  is a *Euclidean* reconstruction, i.e., it differs from the true one by a similarity.

The set of camera matrices can always be transformed to the following canonical form by post-multiplying each  $P_i$  by the matrix  $[P_1; 0 \ 0 \ 0 \ 1]^{-1}$ :

$$P_1 = [I \mid \mathbf{0}] \quad P_i = [Q_i \mid \mathbf{q}_i]. \quad (1)$$

In this situation, the collineation of space  $H$  performing the Euclidean upgrade has the following structure:

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{v}^\top & \lambda \end{bmatrix} \quad (2)$$

where  $K_1$  is the calibration matrix of the first camera,  $\mathbf{v}$  a vector which determines the location of the plane at infinity and  $\lambda$  a scalar fixating the overall scale of the reconstruction.

The technique we are about to describe is based on two stages:

1. Given a guess on the intrinsic parameters of two cameras compute a consistent upgrading collineation. This yields an estimate of all cameras but the first.
2. Score the intrinsic parameters of these  $n - 1$  cameras based on the likelihood of skew, aspect ratio and principal point.

The space of the intrinsic parameters of the two cameras is enumerated and the best solution is eventually refined via non-linear least squares.

### 2.1 Estimation of the Plane at Infinity

In this section we will show how to compute the plane at infinity given two perspective projection matrices and their intrinsic parameters. This procedure is, in a sense, the dual of the second step of the stratified autocalibration [11] in which the intrinsic parameters are recovered given the plane at infinity. This problem has been dealt with for the first time in [12] where it has been turned into a linear least squares system. We shall derive here a closed form solution.

Given two projective cameras

$$P_1 = [I \mid \mathbf{0}] \quad P_2 = [Q_2 \mid \mathbf{q}_2] \tag{3}$$

and their intrinsic parameters matrices  $K_1$  and  $K_2$  respectively, the upgraded, Euclidean versions of the perspective projection matrices are equal to:

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{4}$$

$$P_2^E = K_2 [R_2 \mid \mathbf{t}_2] \simeq P_2 H = [Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top \mid \lambda \mathbf{q}_2] \tag{5}$$

with the symbol  $\simeq$  meaning “equality up to a scale”. The rotation  $R_2$  can therefore be equated to the following:

$$R_2 \simeq K_2^{-1} (Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{6}$$

in which it is expressed as the sum of a 3 by 3 matrix and a rank 1 term.

Using the constraints on orthogonality between rows or columns of a rotation matrix, one can solve for  $\mathbf{v}$  finding the value that makes the right hand side of (6) equal up to a scale to a rotation. The solution can be obtained in closed form by noting that there always exists a rotation matrix  $R^*$  such as:  $R^* \mathbf{t}_2 = [ \|\mathbf{t}_2\| \ 0 \ 0 ]^\top$ . Left multiplying it to (6) yields:

$$R^* R_2 \simeq \overbrace{R^* K_2^{-1} Q_2 K_1}^W + [ \|\mathbf{t}_2\| \ 0 \ 0 ]^\top \mathbf{v}^\top \tag{7}$$

Calling the right hand side first term  $W$  and its rows  $\mathbf{w}_i^\top$ , we arrive at the following:

$$R^* R_2 = \left[ \begin{array}{c} \mathbf{w}_1^\top + \|\mathbf{t}_2\| \mathbf{v}^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \end{array} \right] / \|\mathbf{w}_3\| \tag{8}$$

in which the last two rows are independent from the value of  $\mathbf{v}$  and the correct scale has been recovered normalizing to unit norm each side of the equation.

Since the rows of the right hand side form an orthonormal basis, we can recover the first one taking the cross product of the other two. Vector  $\mathbf{v}$  is therefore equal to:

$$\mathbf{v} = (\mathbf{w}_2 \times \mathbf{w}_3 / \|\mathbf{w}_3\| - \mathbf{w}_1) / \|\mathbf{t}_2\| \tag{9}$$

The upgrading collineation  $H$  can be computed using (2); the term  $\lambda$  can be arbitrarily chosen, as it will just influence the overall scale of the reconstruction. Its sign however will affect the cheirality of the reconstruction, so it must be chosen positive if cheirality was previously adjusted.

## 2.2 Estimation of the Intrinsic Parameters

In the preceding section we showed how to compute the location of the plane at infinity given the calibration parameters of two of the cameras of the projective reconstruction to upgrade. When these calibration parameters are known only approximately, we are not guaranteed anymore that the right hand side of (8) will be a rotation matrix because  $\mathbf{w}_2$  and  $\mathbf{w}_3$  will not be mutually orthogonal, nor have equal, unit norm. However, (9) will still yield the value of  $\mathbf{v}$  that makes the right hand side of (8) closest to a rotation in Frobenius norm. Hence, the derived upgrading collineation  $H$  will produce an *approximate* Euclidean reconstruction.

The autocalibration algorithm we propose consists in enumerating through all possible matrices of intrinsics of two cameras  $K_1$  and  $K_2$  checking whether the entire resulting reconstruction has the desired properties in terms of  $K_2 \dots K_n$ . The process is well-defined, since the search space is naturally bounded by the finiteness of the acquisition devices.

In order to sample the space of calibration parameters we can safely assume, as customary, null skew and unit aspect ratio: this leaves the focal length and the principal point location as free parameters. However, as expected, the value of the plane at infinity is in general far more sensitive to errors in the estimation of focal length values rather than the image center. Thus, we can iterate just over focal lengths  $f_1$  and  $f_2$  assuming the principal point to be centered on the image; the error introduced with this approximation is normally well-within the radius of convergence of the subsequent non-linear optimization. The search space is therefore reduced to a bounded region of  $\mathbb{R}^2$ .

To score each sampled point  $(f_1, f_2)$ , we consider the aspect ratio, skew and principal point location of the resulting transformed camera matrices and aggregate their respective value into a single cost function:

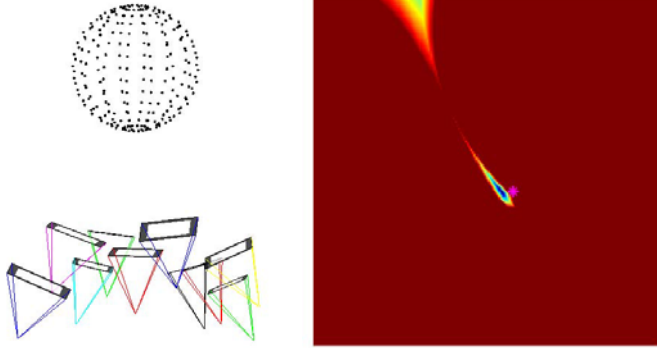
$$\{f_1, f_2\} = \arg \min_{f_1, f_2} \sum_{\ell=2}^n \mathcal{C}^2(K_\ell) \quad (10)$$

where  $K_\ell$  is the intrinsic parameters matrix of the  $\ell$ -th camera after the Euclidean upgrade determined by  $(f_1, f_2)$ , and

$$\mathcal{C}(K) = \overbrace{w_{sk} |k_{1,2}|}^{\text{skew}} + \overbrace{w_{ar} |k_{1,1} - k_{2,2}|}^{\text{aspect ratio}} + \overbrace{w_{u_o} |k_{1,3}| + w_{v_o} |k_{2,3}|}^{\text{principal point}} \quad (11)$$

where  $k_{i,j}$  denotes the entry  $(i, j)$  of  $K$  and  $w$  are suitable weights, computed as in [4]. The first term of (11) takes into account the skew, which is expected to be 0, the second one penalizes cameras with aspect ratio different from 1 and the last two weigh down cameras where the principal point is away from the image centre. If a sufficient (according to the autocalibration ‘‘counting argument’’ [13]) number of cameras is available, the terms related to the principal point can be dropped, thereby leaving it free to move.

As an example, Fig. 1 shows the aggregated cost for a ten camera synthetic dataset, obtained with the aforementioned method. More in detail, Fig. 2 depicts the profiles of each of the term of (11) for two sample cameras. As it can be seen,

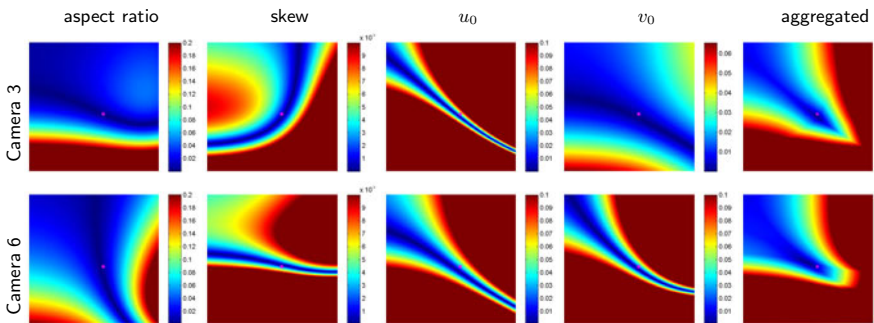


**Fig. 1.** A ten camera synthetic reconstruction and the resulting aggregated cost function. An asterisk marks the correct solution.

the cost profiles have very clear valleys and collectively concur to identify the correct solution, displayed in the graphs as an asterisk.

Even the aggregate cost from just a single camera can still identify a unambiguous minima. This situation is equivalent to the task of identifying the focal lengths of two cameras from their fundamental matrix. This problem, studied extensively in [12, 14, 15], was demonstrated to be essentially ill-conditioned. Our approach is stabler since it structurally requires the solution to be in a valid region of the parameter space. The solution clearly improves as more and more cameras are added.

Finally, the solution selected is refined by non-linear minimization; since it is usually very close to a minima, just a few iterations of a Levenberg-Marquardt solver are necessary for convergence. The employed cost function is the same reported in (10).



**Fig. 2. Cost functions.** The two rows refer to cost functions relative to different cameras of a same dataset. From left to right, are shown the profiles of aspect ratio, skew, principal point  $u_0$  and  $v_0$  coordinates and their aggregated value as function of the focal lengths of the reference cameras. Cooler colors correspond to lower values of the cost function. A asterisk marks the correct solution.

---

**Algorithm 1.** Autocalibration pseudo-code

---

```

input : a set of PPMs  $P$  and their viewpoints  $V$ 
output: their upgraded, euclidean counterparts

1 foreach  $P$  do  $P \leftarrow V^{-1}P/\|P_{3,1:3}\|$  /* normalization */
2 foreach  $K_1, K_2$  do /* iterate over focal pairs */
3   compute  $\Pi_\infty$ 
4   build  $H$  from (2)
5   foreach  $P$  do /* compute cost profiles */
6      $P_E \leftarrow PH$ 
7      $K \leftarrow$  intrinsics of  $P_E$ 
8     compute  $\mathcal{C}(K)$  from (11)
9   end
10 end
11 aggregate cost and select minimum
12 refine non-linearly
13 foreach  $P$  do  $P \leftarrow VPH$  /* de-normalization, upgrade */

```

---

The entire procedure is presented as pseudo-code in Algorithm 1. With the perspective projection matrices the code presented takes as input also the viewpoint matrices of the cameras, defined as:

$$V = \frac{1}{2} \begin{bmatrix} \sqrt{w^2 + h^2} & 0 & w \\ 0 & \sqrt{w^2 + h^2} & h \\ 0 & 0 & 2 \end{bmatrix} \tag{12}$$

where  $w$  and  $h$  are respectively the width and height of each image. This piece of data is used inside the algorithm to normalize camera matrices. While this is not mandatory, we recommend it to improve the numerical conditioning of the algorithm.

The algorithm shows remarkable convergence properties; it has been observed to fail only when the sampling of the focal space was not sufficiently dense (in practice, less than twenty focals in each direction), and therefore all the tested infinity planes were not close enough to the correct one. Such problems are easy to detect, since they usually bring the final, refined solution outside the legal search space.

### 3 Experimental Evaluation

We report here several tests on synthetic and concrete datasets. For the experiments, unless otherwise specified, we sampled the focal space using 50 logarithmically spaced divisions in the range  $[0.3 \dots 3]$ . Please note that, being cameras normalized, a focal length of 1 unit correspond to the length of the image diagonal in pixels.



### 3.1 Synthetic Tests

For this series of tests, we generated several synthetic reconstructions with twenty cameras looking at the unit sphere. Each camera was chosen having different parameters except for skew, which was set equal to zero for all perspective projection matrices. The other characteristics were selected by a random process inside the valid parameter space. The virtual viewport size for each camera was [1024, 768] units, leading to focal lengths and principal points coordinates of comparable magnitude. We built projectively equivalent reconstructions multiplying the Euclidean frame for a random collineation.

**Sampling rate.** The top two graphs of Fig. 3 shows the relationship between the number divisions used in the focal search phase and the error of the resulting autocalibration for focal length and skew respectively, averaged over 100 trials. The focal length error has the form:

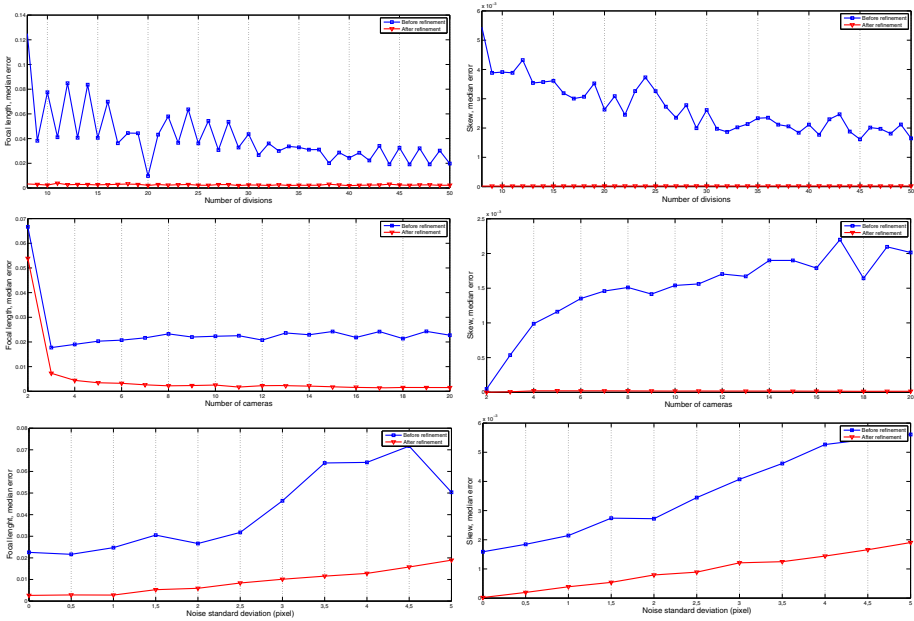
$$\varepsilon = \frac{1}{n} \sum_{\ell=1}^n \Delta f \quad (13)$$

where  $\Delta f$  is defined in equation 14. The error function used for skew has a similar formulation.

For too low rates of sampling, corresponding to the left side of the diagram, the chance of picking a solution close to the correct one is very low. Most of the time the subsequent minimization outputs parameters outside the valid range, generally converging towards the trivial null focal solution. As soon as the focal lengths are sampled with a sufficient degree of accuracy, the residual of the recovered solution becomes and stay low. When this happens, the proposed solution is usually very near to the correct one, and the following non-linear minimization has no problem to converge to the correct, best calibration parameters.

The total elapsed time follows a quadratic law, as expected. At the far right of the diagram, corresponding to fifty divisions for each focal, the total time spent (search plus refinement) is roughly 3 seconds, implemented as a MATLAB script. The omitted graphs for aspect ratio and principal point location show equivalent behaviour.

**Number of cameras.** In this section we verify the stability of the algorithm as the number of cameras varies from two to twenty. For uniformity all reported results were obtained with the full cost function described in (11), even for experiments which, having a sufficient number of cameras, could use fewer constraints. Results reported in the middle graphs of Fig. 3 are averaged over 100 runs of the algorithm. As shown, the algorithm is able to converge to the correct calibration parameters for all but the two-cameras setup, in which it trades focal length accuracy for a lower magnitude of skew. The resulting solution is still very close to the ground truth. From three cameras onwards the method successfully disambiguates the uncertainty.



**Fig. 3. Synthetic tests.** Median autocalibration error  $\varepsilon$  as a function of: the number of sampling divisions (top), the number of cameras (middle), the standard deviation of noise for both focal length (left) and skew (right).

**Noise resilience.** Our final synthetic test verifies the resilience to noise; several reconstructions were built from the ground truth perturbing the point projections with Gaussian noise and recovering each camera by DLT based resection [16]. The bottom plots of Fig. 3 shows the dependency of the error  $\varepsilon$  on the standard deviation of the added noise. Again, the results were averaged over 100 runs of the algorithm. As it can be seen the method is fairly stable, degrading quite gracefully as the standard deviation of noise increases.

Again, omitted graphs for aspect ratio and principal point location behave similarly.

### 3.2 Comparative Tests

We compare our approach to a classical, linear technique based on the DIAQ constraints and a recent stratified method based on least squares minimization of the modulus constraint embedded in a branch and bound framework.

The first algorithm is our implementation of the iterative dual linear auto-calibration algorithm described in [5], modified to use the weights of [4] and to enforce at every iteration the positive (negative) semi-definiteness of the DIAQ. As explained in [17], the closest semi-definite approximation of a matrix in Frobenius norm can be obtained, assuming a single offending value, zeroing the eigenvalue with sign different from the others. This can be easily done during the rank 3 approximation step of the original algorithm. Several informal tests, not

**Table 1.** Comparison of results obtained on the dataset from [6]

Algorithm	Cameras	$\Delta f$	Success rate	Time
Dual linear	5	5.4012e-2	57%	0.39
	10	2.6522e-3	84 %	0.45
	20	1.5433e-3	90 %	0.78
DL + QA upgrade	5	2.7420e-2	63 %	0.41
	10	1.8943e-3	83 %	0.43
	20	1.1295e-3	92 %	0.68
Chandraker <i>et al</i> [9]	5	9.9611e-3	100 %	584.12
	10	4.7925e-3	100 %	560.56
	20	1.0461e-3	100 %	602.32
Our method	5	2.7546e-3	100 %	0.35
	10	1.3005e-3	100 %	0.72
	20	8.2266e-4	100 %	1.62

reported here, demonstrated this algorithm to have better convergence properties of both its parents [54]. We report also the results obtained by this method when coupled with the preliminary quasi-affine upgrade step detailed in [18].

The second method we compare to is the algorithm described in [9], a stratified autocalibration approach based on a branch and bound framework using convex relaxations minimizations. We tested the implementation of the authors (available from <http://vision.ucsd.edu/stratum/>), coupled with the SeDuMi [19] library version 1.1R3 which was used in the original article (the latest version 1.21 is not compatible with the code) under MATLAB R2009a.

The synthetic test dataset, the same used in [9], is composed of twenty projective cameras and points, with known ground truth and Gaussian noise of standard deviation  $\sigma$  added to image coordinates. We report results obtained by our and the aforementioned methods over a hundred trials in the case of  $\sigma = 0.1\%$  using the same metric defined in the original article:

$$\Delta f = \left| \frac{f_x + f_y}{f_x^{GT} + f_y^{GT}} - 1 \right| \quad (14)$$

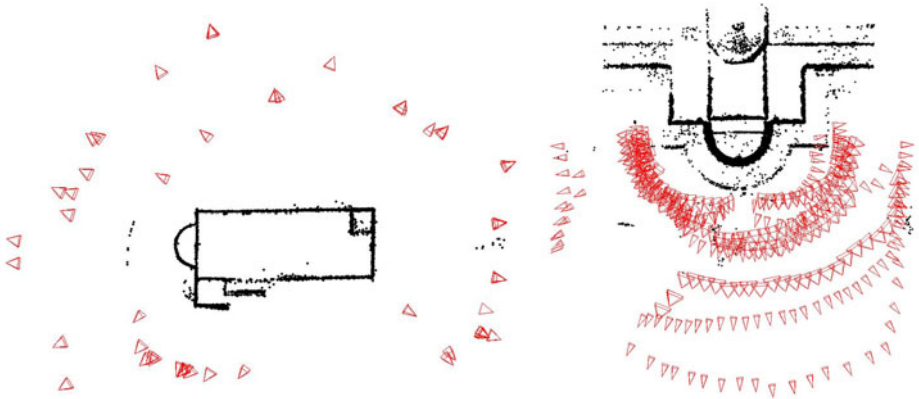
where  $f_x$  and  $f_y$  are the focal entries of the calibration matrix and  $f_x^{GT}$  and  $f_y^{GT}$  the respective ground truth values. Results are reported in Tab. 1. The linear algorithm, which we pick as baseline case, achieves good results in terms of  $\Delta f$  but shows poor convergence properties, especially for lower number of cameras. Similar numerical results are unsurprisingly obtained coupling the method with the quasi-affine upgrade of [18], with slightly higher percentages of success. Both the algorithm described in [9] and our method never failed on this dataset, with a slight numerical advantage of our proposal.

### 3.3 Real World Example

We finally tested our algorithm on two real reconstructions, *Pozzoveggiani* and *Duomo*, composed respectively of 52 and 333 cameras (data available from

**Table 2.** Comparison of results obtained on real reconstructions

Algorithm	<i>Pozzoveggiani</i>		<i>Duomo</i>	
	$\Delta f$	Succ. rate	$\Delta f$	Succ. rate
Dual linear	3.0815e-2	19 %	9.3255e-2	8 %
DL + QA upgrade	8.9261e-3	22 %	7.6403e-2	13 %
Our method	3.9733e-3	100 %	2.9293e-3	100 %



**Fig. 4.** *Pozzoveggiani* (left) and *Duomo* (right) reconstruction after the upgrade found by our method

<http://profs.sci.univr.it/~fusiello/demo/samantha/>). These reconstructions, refined through bundle adjustment, have relatively low noise levels and were used as ground truth for the subsequent tests. Again, a total of a hundred trials were conducted for each set, multiplying the projective reconstructions for a random collineation while discarding the ones with very low condition number. In our method we also picked at random the reference views to be used for the estimation of the plane at infinity.

Results are reported in Tab. 2. With respect to the synthetic case, we can note a substantial decrease of the success rate of both linear algorithms which was instead expected to increase with the number of cameras. An informal audit of the code showed the effect to be caused both by noise and by the larger number of iterations required for convergence, which in turn increase the chance of encountering a failure case.

Algorithm 9 is missing from Tab. 2 because we were not able to obtain valid solutions on these data, even by varying the tolerance  $\epsilon$  and the maximal number of iterations for both the affine and metric upgrade steps.

Our approach achieves on both datasets flawless success rate. Instances of the upgraded reconstructions can be qualitatively evaluated in Fig. 4.

## 4 Conclusions

We presented a practical autocalibration algorithm showing results comparable to the state of the art. Our approach is fast, easy to implement and shows remarkable convergence properties.

Future research will be aimed at developing sub-linear search strategies in the space of calibration parameters, which is made possible by the structure of the cost profiles.

## Acknowledgments

The use of code and data from [\[6\]](#) is gratefully acknowledged.

## References

1. Maybank, S.J., Faugeras, O.: A theory of self-calibration of a moving camera. *International Journal of Computer Vision* 8, 123–151 (1992)
2. Triggs, B.: Autocalibration and the absolute quadric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pp. 609–614 (1997)
3. Pollefeys, M., Koch, R., Van Gool, L.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: *Proceedings of the International Conference on Computer Vision, Bombay*, pp. 90–95 (1998)
4. Pollefeys, M., Verbiest, F., Van Gool, L.: Surviving dominant planes in uncalibrated structure and motion recovery. In: *Proceedings of the European Conference on Computer Vision*, pp. 837–851 (2002)
5. Seo, Y., Heyden, A., Cipolla, R.: A linear iterative method for auto-calibration using the dac equation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 880 (2001)
6. Chandraker, M., Agarwal, S., Kahl, F., Nister, D., Kriegman, D.: Autocalibration via rank-constrained estimation of the absolute quadric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
7. Fusiello, A., Benedetti, A., Farenzena, M., Busti, A.: Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1633–1638 (2004)
8. Bocquillon, B., Bartoli, A., Gurdjos, P., Crouzil, A.: On constant focal length self-calibration from multiple views. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007)
9. Chandraker, M., Agarwal, S., Kriegman, D., Belongie, S.: Globally optimal affine and metric upgrades in stratified autocalibration. In: *Proceedings of the International Conference on Computer Vision*, pp. 1–8 (2007)
10. Hartley, R., Hayman, E., de Agapito, L., Reid, I.: Camera calibration and the search for infinity. In: *Proceedings of the International Conference on Computer Vision* (1999)
11. Faugeras, O.: Stratification of 3-D vision: projective, affine, and metric representations. *Journal of the Optical Society of America A* 12, 465–484 (1994)
12. Bougnoux, S.: From projective to Euclidean space under any practical situation, a criticism of self-calibration. In: *Proceedings of the International Conference on Computer Vision, Bombay*, pp. 790–796 (1998)

13. Luong, Q.T., Viéville, T.: Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding* 64, 193–229 (1996)
14. Sturm, P.: On focal length calibration from two views. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, USA, vol. II*, pp. 145–150 (2001)
15. Newsam, G.N., Huynh, D.Q., Brooks, M.J., p. Pan, H.: Recovering unknown focal lengths in self-calibration: An essentially linear algorithm and degenerate configurations. *Int. Arch. Photogrammetry & Remote Sensing*, 575–580 (1996)
16. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2003)
17. Higham, N.J.: Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications* 103, 103–118 (1988)
18. Hartley, R.I.: Chirality. *Int. J. Comput. Vision* 26, 41–61 (1998)
19. Sturm, J.F.: Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software* 11-12, 625–653 (1999)

# Author Index

- Abugharbieh, Rafeef IV-651  
Adler, Amir II-622  
Aeschliman, Chad II-594  
Agapito, Lourdes II-15, IV-283,  
IV-297  
Agarwal, Sameer II-29  
Agrawal, Amit I-100, II-237, III-129  
Ahuja, Narendra II-223, IV-87,  
VI-393, V-644  
Ai, Haizhou VI-238  
Alahari, Karteek IV-424  
Albarelli, Andrea V-519  
Alexe, Bogdan IV-452, V-380  
Aloimonos, Y. II-506  
Aloimonos, Yiannis V-127  
Alpert, Sharon IV-750  
Alterovitz, Ron III-101  
Andriyenko, Anton I-466  
Angst, Roland III-144  
Appia, Vikram I-73, VI-71  
Arbelaez, Pablo IV-694  
Arora, Chetan III-552  
Arras, Kai O. V-296  
Åström, Kalle II-114  
Avidan, Shai V-268  
Avraham, Tamar V-99  
Ayazoglu, Mustafa II-71
- Baatz, Georges VI-266  
Babenko, Boris IV-438  
Bae, Egil VI-379  
Bagdanov, Andrew D. VI-280  
Bagnell, J. Andrew VI-57  
Bai, Jiamin II-294  
Bai, Xiang III-328, V-15  
Bai, Xue V-617  
Bajcsy, Ruzena III-101  
Baker, Simon I-243  
Balikai, Anupriya IV-694  
Banerjee, Subhashis III-552  
Bao, Hujun V-422  
Baraniuk, Richard G. I-129  
Bar-Hillel, Aharon IV-127  
Barinova, Olga II-57
- Barnes, Connelly III-29  
Barreto, João P. IV-382  
Bartoli, Adrien II-15  
Basri, Ronen IV-750  
Baust, Maximilian III-580  
Behmo, Régis IV-171  
Belongie, Serge I-591, IV-438  
Ben, Shenglan IV-44  
BenAbdelkader, Chiraz VI-518  
Ben-Ezra, Moshe I-59  
Berg, Alexander C. I-663, V-71  
Berg, Tamara L. I-663  
Bernal, Hector I-762  
Bhakta, Vikrant VI-405  
Bischof, Horst III-776, VI-29, V-29  
Bitsakos, K. II-506  
Bizheva, Kostadinka K. III-44  
Black, Michael J. I-285  
Blanz, Volker I-299  
Boben, Marko V-687  
Boley, Daniel IV-722  
Boltz, Sylvain III-692  
Boucher, Jean-Marc IV-185, IV-764  
Boult, Terrance III-481  
Bourdev, Lubomir VI-168  
Bowden, Richard VI-154  
Boyer, Edmund IV-326  
Boykov, Yuri VI-379, V-211  
Bradski, Gary V-658  
Brandt, Jonathan VI-294  
Brandt, Sami S. IV-666  
Branson, Steve IV-438  
Bregler, Christoph VI-140  
Breitenreicher, Dirk II-494  
Brendel, William II-721  
Breuer, Pia I-299  
Bronstein, Alex III-398  
Bronstein, Alexander M. II-197  
Bronstein, Michael II-197, III-398  
Brown, Michael S. VI-323  
Brox, Thomas I-438, VI-168, V-282  
Bruhn, Andrés IV-568  
Bu, Jiajun V-631  
Burgoon, Judee K. VI-462

- Burschka, Darius II-183  
 Byröd, Martin II-114
- Cagniard, Cedric IV-326  
 Cai, Qin III-229  
 Calonder, Michael IV-778  
 Camps, Octavia II-71  
 Cannons, Kevin J. IV-511  
 Cao, Yang V-729  
 Caplier, Alice I-313  
 Castellani, Umberto VI-15  
 Chandraker, Manmohan II-294  
 Chao, Hongyang III-342  
 Charpiat, Guillaume V-715  
 Chaudhry, Rizwan II-735  
 Chellappa, Rama I-129, III-286, V-547  
 Chen, Chih-Wei II-392  
 Chen, Chun V-631  
 Chen, David VI-266  
 Chen, Jiansheng IV-44  
 Chen, Jiun-Hung III-621  
 Chen, Siqi V-715  
 Chen, Weiping III-496  
 Chen, Xiaowu IV-101  
 Chen, Xilin I-327, II-308  
 Chen, Yu III-300  
 Chen, Yuanhao V-43  
 Cheong, Loong-Fah III-748  
 Chia, Liang-Tien I-706, IV-1  
 Chin, Tat-Jun V-533  
 Cho, Minsu V-492  
 Choi, Wongun IV-553  
 Christensen, Marc VI-405  
 Chua, Tat-Seng IV-30  
 Chum, Ondřej III-1  
 Chung, Albert C.S. III-720  
 Cipolla, Roberto III-300  
 Clausi, David A. III-44  
 Clipp, Brian IV-368  
 Cohen, Laurent D. V-771  
 Cohen, Michael I-171  
 Collins, Robert T. V-324  
 Collins, Roderic I-549, II-664  
 Courchay, Jérôme II-85  
 Cremers, Daniel III-538, V-225  
 Criminisi, Antonio III-510  
 Cristani, Marco II-378, VI-15  
 Cucchiara, Rita VI-196  
 Curless, Brian I-171, VI-364
- Dai, Shengyang I-480  
 Dai, Yuchao IV-396  
 Dalalyan, Arnak II-85, IV-171  
 Dammertz, Holger V-464  
 Darrell, Trevor I-677, IV-213  
 Davies, Ian III-510  
 Davis, Larry S. II-693, IV-199, VI-476  
 Davison, Andrew J. III-73  
 De la Torre, Fernando II-364  
 Del Bue, Alessio III-87, IV-283, IV-297  
 Deng, Jia V-71  
 Deselaers, Thomas IV-452, V-380  
 Di, Huijun IV-525  
 Dickinson, Sven II-480, V-183, V-603  
 Dilsizian, Mark VI-462  
 Ding, Chris III-762, IV-793, VI-126  
 Ding, Lei IV-410  
 Ding, Yuanyuan I-15  
 Di Stefano, Luigi III-356  
 Dodgson, Neil A. III-510  
 Domokos, Csaba II-777  
 Dong, Zilong V-422  
 Donoser, Michael V-29  
 Douze, Matthijs I-522  
 Dragon, Ralf II-128  
 Duan, Genquan VI-238  
 Dunn, Enrique IV-368
- Ebert, Sandra I-720  
 Efros, Alexei A. II-322, IV-482  
 Eichel, Justin A. III-44  
 Eichner, Marcin I-228  
 Elad, Michael II-622  
 Elmoataz, Abderrahim IV-638  
 Endres, Ian V-575  
 Eskin, Yulia V-183  
 Ess, Andreas I-397, I-452
- Fablet, Ronan IV-185, IV-764  
 Fan, Jialue I-411, I-480  
 Fang, Tian II-1  
 Farenzena, Michela II-378  
 Farhadi, Ali IV-15  
 Fayad, João IV-297  
 Fazly, Afsaneh V-183  
 Fei-Fei, Li II-392, V-71, V-785  
 Fergus, Rob I-762, VI-140  
 Fermüller, C. II-506  
 Fernández, Carles II-678  
 Ferrari, Vittorio I-228, IV-452, V-380



- Fidler, Sanja V-687  
 Fieguth, Paul W. III-44  
 Finckh, Manuel V-464  
 Finkelstein, Adam III-29  
 Fite-Georgel, Pierre IV-368  
 Fitzgibbon, Andrew I-776  
 Fleet, David J. III-243  
 Flint, Alex V-394  
 Forsyth, David IV-15, IV-227,  
 VI-224, V-169  
 Fowlkes, Charless IV-241  
 Frahm, Jan-Michael II-142, IV-368  
 Franke, Uwe IV-582  
 Fraundorfer, Friedrich IV-269  
 Freeman, William T. III-706  
 Freifeld, Oren I-285  
 Fritz, Mario IV-213  
 Fua, Pascal III-58, III-370,  
 III-635, IV-778  
 Fuh, Chiou-Shann VI-84  
 Fusiello, Andrea I-790, V-589  
  
 Gall, Juergen I-620, III-425  
 Gallagher, Andrew V-169  
 Gallup, David III-229, IV-368  
 Galun, Meirav IV-750  
 Gammeter, Stephan I-734  
 Gao, Shenghua IV-1  
 Gao, Wen I-327, II-308  
 Gao, Yongsheng III-496  
 Ge, Weina V-324  
 Gehler, Peter I-143, VI-98  
 Ghanem, Bernard II-223  
 Gherardi, Riccardo I-790  
 Glocker, Ben III-272  
 Godec, Martin III-776  
 Goldberg, Chen IV-127  
 Goldluecke, Bastian V-225  
 Goldman, Dan B. III-29  
 Gong, Leiguang IV-624  
 Gong, Yihong VI-434  
 González, Jordi II-678, VI-280  
 Gopalan, Raghuraman III-286  
 Gould, Stephen II-435, IV-497, V-338  
 Grabner, Helmut I-369  
 Gray, Douglas VI-434  
 Gryn, Jacob M. IV-511  
 Grzeszczuk, Radek VI-266  
 Gu, Chunhui V-408  
 Gu, Steve III-663  
 Gu, Xianfeng V-672  
 Gualdi, Giovanni VI-196  
 Guan, Peng I-285  
 Guillaumin, Matthieu I-634  
 Guo, Huimin VI-476  
 Guo, Yanwen III-258  
 Gupta, Abhinav IV-199, IV-482  
 Gupta, Ankit I-171  
 Gupta, Mohit I-100  
  
 Hager, Gregory D. II-183  
 Hall, Peter IV-694  
 Hamarneh, Ghassan IV-651  
 Han, Hu II-308  
 Han, Mei II-156  
 Han, Tony X. III-200, III-748  
 Harada, Tatsuya IV-736  
 Hartley, Richard III-524  
 Hauberg, Søren I-425, VI-43  
 Havlena, Michal II-100  
 He, Jinping IV-44  
 He, Kaiming I-1  
 He, Mingyi IV-396  
 He, Xuming IV-539  
 Hebert, Martial I-508, I-536,  
 IV-482, VI-57  
 Hedau, Varsha VI-224  
 Heibel, T. Hauke III-272  
 Heikkilä, Janne I-327, V-366  
 Hejrati, Mohsen IV-15  
 Hel-Or, Yacov II-622  
 Hesch, Joel A. IV-311  
 Hidane, Moncef IV-638  
 Hinton, Geoffrey E. VI-210  
 Hirzinger, Gerhard II-183  
 Hockenmaier, Julia IV-15  
 Hoiem, Derek VI-224, V-575  
 Hoogs, Anthony I-549, II-664  
 Horaud, Radu V-743  
 Horbert, Esther I-397  
 Hou, Tingbo III-384  
 Hsu, Gee-Sern I-271  
 Hu, Yiqun I-706  
 Hua, Gang I-243, III-200  
 Huang, Chang I-383, III-314  
 Huang, Dong II-364  
 Huang, Heng III-762, IV-793, VI-126  
 Huang, Junzhou III-607, IV-624  
 Huang, Thomas S. III-566, VI-490,  
 V-113, V-141

- Hung, Yi-Ping I-271  
 Huttenlocher, Daniel P. II-791  
  
 Idrees, Haroon III-186  
 Ik Cho, Nam II-421  
 Ikizler-Cinbis, Nazli I-494  
 Ilic, Slobodan IV-326  
 Ilstrup, David I-200  
 Ip, Horace H.S. VI-1  
 Isard, Michael I-648, III-677  
 Ishiguro, Hiroshi VI-337  
 Ito, Satoshi II-209, V-701  
 Ivanov, Yuri II-735  
  
 Jain, Arpit IV-199  
 Jamieson, Michael V-183  
 Jen, Yi-Hung IV-368  
 Jégou, Hervé I-522  
 Jeng, Ting-Yueh I-605  
 Ji, Qiang VI-532  
 Jia, Jiaya I-157, V-422  
 Jiang, Lin VI-504  
 Jiang, Xiaoyue IV-58  
 Jin, Xin IV-101  
 Johnson, Micah K. I-31  
 Johnson, Tim IV-368  
 Jojic, Nebojsa VI-15  
 Joshi, Neel I-171  
 Jung, Kyomin II-535  
 Jung, Miyouon I-185  
  
 Kak, Avinash C. II-594  
 Kalra, Prem III-552  
 Kankanhalli, Mohan IV-30  
 Kannala, Juho V-366  
 Kapoor, Ashish I-243  
 Kappes, Jörg Hendrik III-735  
 Kato, Zoltan II-777  
 Katti, Harish IV-30  
 Ke, Qifa I-648  
 Kembhavi, Aniruddha II-693  
 Kemelmacher-Shlizerman, Ira I-341  
 Keriven, Renaud II-85  
 Keutzer, Kurt I-438  
 Khuwuthyakorn, Pattaraporn II-636  
 Kim, Gunhee V-85  
 Kim, Hyeongwoo I-59  
 Kim, Jaewon I-86  
 Kim, Minyoung III-649  
 Kim, Seon Joo VI-323  
  
 Kim, Tae-Kyun III-300  
 Knopp, Jan I-748  
 Kohli, Pushmeet II-57, II-535,  
 III-272, V-239  
 Kohno, Tadayoshi VI-364  
 Kokkinos, Iasonas II-650  
 Kolev, Kalin III-538  
 Koller, Daphne II-435, IV-497, V-338  
 Kolmogorov, Vladimir II-465  
 Komodakis, Nikos II-520  
 Koo, Hyung Il II-421  
 Köser, Kevin VI-266  
 Krömer, Oliver II-566  
 Krupka, Eyal IV-127  
 Kubota, Susumu II-209, V-701  
 Kulikowski, Casimir IV-624  
 Kulis, Brian IV-213  
 Kuniyoshi, Yasuo IV-736  
 Kuo, Cheng-Hao I-383  
 Kwatra, Vivek II-156  
  
 Ladický, L'ubor IV-424, V-239  
 Lalonde, Jean-François II-322  
 Lampert, Christoph H. II-566, VI-98  
 Lanman, Douglas I-86  
 Lao, Shihong VI-238  
 Larlus, Diane I-720  
 Latecki, Longin Jan III-411, V-450,  
 V-757  
 Lauze, François VI-43  
 Law, Max W.K. III-720  
 Lawrence Zitnick, C. I-171  
 Lazarov, Maxim IV-72  
 Lazebnik, Svetlana IV-368, V-352  
 LeCun, Yann VI-140  
 Lee, David C. I-648  
 Lee, Jungmin V-492  
 Lee, Kyoung Mu V-492  
 Lee, Ping-Han I-271  
 Lee, Sang Wook IV-115  
 Lefort, Riwal IV-185  
 Leibe, Bastian I-397  
 Leistner, Christian III-776, VI-29  
 Lellmann, Jan II-494  
 Lempitsky, Victor II-57  
 Lensch, Hendrik P.A. V-464  
 Leonardis, Aleš V-687  
 Lepetit, Vincent III-58, IV-778  
 Levi, Dan IV-127  
 Levin, Anat I-214

- Levinshtein, Alex II-480  
Lewandowski, Michał VI-547  
Lézoray, Olivier IV-638  
Li, Ang III-258  
Li, Chuan IV-694  
Li, Hanxi II-608  
Li, Hongdong IV-396  
Li, Kai V-71  
Li, Na V-631  
Li, Ruonan V-547  
Li, Yi VI-504  
Li, Yin III-790  
Li, Yunpeng II-791  
Li, Zhiwei IV-157  
Lian, Wei V-506  
Lian, Xiao-Chen IV-157  
Lim, Yongsub II-535  
Lin, Dahua I-243  
Lin, Liang III-342  
Lin, Yen-Yu VI-84  
Lin, Zhe VI-294  
Lin, Zhouchen I-115, VI-490  
Lindenbaum, Michael V-99  
Ling, Haibin III-411  
Liu, Baiyang IV-624  
Liu, Ce III-706  
Liu, Jun VI-504  
Liu, Risheng I-115  
Liu, Shuaicheng VI-323  
Liu, Siying II-280  
Liu, Tyng-Luh VI-84  
Liu, Wenyu III-328, V-15  
Liu, Xiaoming I-354  
Liu, Xinyang III-594  
Liu, Xiuwen III-594  
Liu, Yazhou I-327  
Livne, Micha III-243  
Lobaton, Edgar III-101  
Lourakis, Manolis I.A. II-43  
Lovegrove, Steven III-73  
Lu, Bao-Liang IV-157  
Lu, Zhiwu VI-1  
Lucey, Simon III-467  
Lui, Lok Ming V-672  
Luo, Jiebo V-169  
Luo, Ping III-342  
  
Ma, Tianyang V-450  
Maheshwari, S.N. III-552  
Mair, Elmar II-183  
Maire, Michael II-450  
Maji, Subhransu VI-168  
Majumder, Aditi IV-72  
Makadia, Ameesh V-310  
Makris, Dimitrios VI-547  
Malik, Jitendra VI-168, V-282  
Manduchi, Roberto I-200  
Mansfield, Alex I-143  
Marcombes, Paul IV-171  
Mario Christoudias, C. I-677  
Marks, Tim K. V-436  
Matas, Jiří III-1  
Matikainen, Pyry I-508  
Matsushita, Yasuyuki II-280  
Matthews, Iain III-158  
McCloskey, Scott I-15, VI-309  
Meer, Peter IV-624  
Mehrani, Paria V-211  
Mehran, Ramin III-439  
Mei, Christopher V-394  
Mensink, Thomas IV-143  
Metaxas, Dimitris III-607, VI-462  
Michael, Nicholas VI-462  
Micheals, Ross III-481  
Mikami, Dan III-215  
Mikulík, Andrej III-1  
Miller, Eric V-268  
Mio, Washington III-594  
Mirmehdi, Majid IV-680, V-478  
Mitra, Niloy J. III-398  
Mitzel, Dennis I-397  
Mnih, Volodymyr VI-210  
Monroy, Antonio V-197  
Montoliu, Raúl IV-680  
Moore, Brian E. III-439  
Moorthy, Anush K. V-1  
Morellas, Vassilios IV-722  
Moreno-Noguer, Francesc III-58,  
III-370  
Mori, Greg II-580, V-155  
Morioka, Nobuyuki I-692  
Moses, Yael III-15  
Mourikis, Anastasios I. IV-311  
Mu, Yadong III-748  
Mukaigawa, Yasuhiro I-86  
Müller, Thomas IV-582  
Munoz, Daniel VI-57  
Murino, Vittorio II-378, VI-15  
Murray, David V-394

- Nadler, Boaz IV-750  
 Nagahara, Hajime VI-337  
 Nakayama, Hideki IV-736  
 Narasimhan, Srinivasa G. I-100, II-322  
 Nascimento, Jacinto C. III-172  
 Navab, Nassir III-272, III-580  
 Nayar, Shree K. VI-337  
 Nebel, Jean-Christophe VI-547  
 Neumann, Ulrich III-115  
 Nevatia, Ram I-383, III-314  
 Ng, Tian-Tsong II-280, II-294  
 Nguyen, Huu-Giao IV-764  
 Niebles, Juan Carlos II-392  
 Nielsen, Frank III-692  
 Nielsen, Mads IV-666, VI-43  
 Nishino, Ko II-763  
 Nowozin, Sebastian VI-98  
 Nunes, Urbano IV-382
- Obrador, Pere V-1  
 Oh, Sangmin I-549  
 Oliver, Nuria V-1  
 Ommer, Björn V-197  
 Orr, Douglas III-510  
 Ostermann, Joern II-128  
 Otsuka, Kazuhiro III-215  
 Oxholm, Geoffrey II-763  
 Özuysal, Mustafa III-58, III-635
- Packer, Ben V-338  
 Pajdla, Tomas I-748  
 Pajdla, Tomáš II-100  
 Paladini, Marco II-15, IV-283  
 Pantic, Maja II-350  
 Papamichalis, Panos VI-405  
 Papanikolopoulos, Nikolaos IV-722  
 Paris, Sylvain I-31  
 Park, Dennis IV-241  
 Park, Hyun Soo III-158  
 Park, Johnny II-594  
 Patel, Ankur VI-112  
 Patras, Ioannis II-350  
 Patterson, Donald IV-610  
 Pätz, Torben V-254  
 Pavlovic, Vladimir III-649  
 Payet, Nadia V-57  
 Pedersen, Kim Steenstrup I-425  
 Pedersoli, Marco VI-280  
 Pele, Ofir II-749  
 Pellegrini, Stefano I-452
- Perdoch, Michal III-1  
 Pérez, Patrick I-522  
 Perina, Alessandro VI-15  
 Perona, Pietro IV-438  
 Perronnin, Florent IV-143  
 Petersen, Kersten IV-666  
 Peyré, Gabriel V-771  
 Pfister, Hanspeter II-251, V-268  
 Philbin, James III-677  
 Pietikainen, Matti I-327  
 Pock, Thomas III-538  
 Polak, Simon II-336  
 Pollefeys, Marc II-142, III-144, IV-269,  
 IV-354, IV-368, VI-266  
 Porta, Josep M. III-370  
 Prati, Andrea VI-196  
 Preusser, Tobias V-254  
 Prinnet, Véronique IV-171  
 Pu, Jian I-257  
 Pugeault, Nicolas VI-154  
 Pundik, Dmitry III-15
- Qin, Hong III-384  
 Qing, Laiyun II-308  
 Quack, Till I-734  
 Quan, Long II-1, V-561
- Rabe, Clemens IV-582  
 Rabin, Julien V-771  
 Radke, Richard J. V-715  
 Raguram, Rahul IV-368  
 Rahtu, Esa V-366  
 Ramalingam, Srikumar III-129, V-436  
 Ramamoorthi, Ravi II-294  
 Ramanan, Deva IV-241, IV-610  
 Ramanathan, Subramanian IV-30  
 Rangarajan, Prasanna VI-405  
 Ranjbar, Mani II-580  
 Rao, Josna IV-651  
 Raptis, Michalis I-577  
 Rashtchian, Cyrus IV-15  
 Raskar, Ramesh I-86  
 Razavi, Nima I-620  
 Reid, Ian V-394  
 Reilly, Vladimir III-186, VI-252  
 Ren, Xiaofeng V-408  
 Resmerita, Elena I-185  
 Richardt, Christian III-510  
 Riemenschneider, Hayko V-29  
 Robles-Kelly, Antonio II-636

- Roca, Xavier II-678  
 Rocha, Anderson III-481  
 Rodolà, Emanuele V-519  
 Rodrigues, Rui IV-382  
 Romeiro, Fabiano I-45  
 Rosenhahn, Bodo II-128  
 Roshan Zamir, Amir IV-255  
 Roth, Stefan IV-467  
 Rother, Carsten I-143, II-465, III-272  
 Roumeliotis, Stergios I. IV-311  
 Roy-Chowdhury, Amit K. I-605  
 Rudovic, Ognjen II-350  
 Russell, Chris IV-424, V-239  
  
 Sadeghi, Mohammad Amin IV-15  
 Saenko, Kate IV-213  
 Saffari, Amir III-776, VI-29  
 Sajadi, Behzad IV-72  
 Sala, Pablo V-603  
 Salo, Mikko V-366  
 Salti, Samuele III-356  
 Salzmann, Mathieu I-677  
 Sánchez, Jorge IV-143  
 Sankar, Aditya I-341  
 Sankaranarayanan, Aswin C. I-129,  
 II-237  
 Sapiro, Guillermo V-617  
 Sapp, Benjamin II-406  
 Satkin, Scott I-536  
 Satoh, Shin'ichi I-692  
 Savarese, Silvio IV-553, V-658  
 Scharr, Hanno IV-596  
 Scheirer, Walter III-481  
 Schiele, Bernt I-720, IV-467, VI-182  
 Schindler, Konrad I-466, IV-467, VI-182  
 Schmid, Cordelia I-522, I-634  
 Schmidt, Stefan III-735  
 Schnörr, Christoph II-494, III-735  
 Schofield, Andrew J. IV-58  
 Schroff, Florian IV-438  
 Schuchert, Tobias IV-596  
 Schwartz, William Robson VI-476  
 Sclaroff, Stan I-494, III-453  
 Sebe, Nicu IV-30  
 Seitz, Steven M. I-341, II-29  
 Seo, Yongduek IV-115  
 Serradell, Eduard III-58  
 Shah, Mubarak III-186, III-439,  
 IV-255, VI-252  
  
 Shan, Qi VI-364  
 Shan, Shiguang I-327, II-308  
 Shapiro, Linda G. III-621  
 Sharma, Avinash V-743  
 Shashua, Amnon II-336  
 Shechtman, Eli I-341, III-29  
 Sheikh, Yaser III-158  
 Shen, Chunhua II-608  
 Shen, Xiaohui I-411  
 Shetty, Sanketh V-644  
 Shi, Yonggang III-594  
 Shih, Jonathan I-663  
 Shiratori, Takaaki III-158  
 Shoaib, Muhammad II-128  
 Shu, Xianbiao VI-393  
 Siegwart, Roland V-296  
 Sigal, Leonid III-243  
 Silva, Jorge G. III-172  
 Singh, Vivek Kumar III-314  
 Sivalingam, Ravishankar IV-722  
 Sivic, Josef I-748, III-677  
 Sminchisescu, Cristian II-480  
 Smith, William A.P. VI-112  
 Snavely, Noah II-29, II-791  
 Soatto, Stefano I-577, III-692  
 Solmaz, Berkan VI-252  
 Sommer, Stefan I-425, VI-43  
 Song, Bi I-605  
 Song, Mingli V-631  
 Song, Yi-Zhe IV-694  
 Spera, Mauro II-378  
 Spinello, Luciano V-296  
 Stalder, Severin I-369  
 Staudt, Elliot I-605  
 Stevenson, Suzanne V-183  
 Stoll, Carsten IV-568  
 Strecha, Christoph IV-778  
 Sturgess, Paul IV-424  
 Sturm, Peter II-85  
 Su, Guangda IV-44  
 Su, Zhixun I-115  
 Sukthankar, Rahul I-508  
 Sun, Jian I-1  
 Sun, Ju III-748  
 Sun, Min V-658  
 Sundaram, Narayanan I-438  
 Sunkavalli, Kalyan II-251  
 Suppa, Michael II-183  
 Suter, David V-533  
 Szeliski, Richard II-29

- Sznaier, Mario II-71  
 Szummer, Martin I-776
- Ta, Vinh-Thong IV-638  
 Taguchi, Yuichi III-129, V-436  
 Tai, Xue-Cheng VI-379  
 Tai, Yu-Wing VI-323  
 Takahashi, Keita IV-340  
 Tan, Ping II-265  
 Tan, Xiaoyang VI-504  
 Tang, Feng III-258  
 Tang, Hao VI-490  
 Tang, Xiaou I-1, VI-420  
 Tanskanen, Petri IV-269  
 Tao, Hai III-258  
 Tao, Linmi IV-525  
 Tao, Michael W. I-31  
 Taskar, Ben II-406  
 Taylor, Graham W. VI-140  
 Theobalt, Christian IV-568  
 Thompson, Paul III-594  
 Tian, Tai-Peng III-453  
 Tighe, Joseph V-352  
 Tingdahl, David I-734  
 Todorovic, Sinisa II-721, V-57  
 Toldo, Roberto V-589  
 Tomasi, Carlo III-663  
 Tombari, Federico III-356  
 Tong, Yan I-354  
 Torii, Akihiko II-100  
 Torr, Philip H.S. IV-424, V-239  
 Torralba, Antonio I-762, II-707, V-85  
 Torresani, Lorenzo I-776  
 Torsello, Andrea V-519  
 Tosato, Diego II-378  
 Toshev, Alexander II-406  
 Tran, Duan IV-227  
 Traver, V. Javier IV-680  
 Tretiak, Elena II-57  
 Triebel, Rudolph V-296  
 Troje, Nikolaus F. III-243  
 Tsang, Ivor Wai-Hung IV-1  
 Tu, Peter H. I-354  
 Tu, Zhuowen III-328, V-15  
 Turaga, Pavan III-286  
 Turaga, Pavan K. I-129  
 Turek, Matthew I-549  
 Turek, Matthew W. II-664  
 Tuzel, Oncel II-237, V-436
- Urtasun, Raquel I-677
- Valgaerts, Levi IV-568  
 Valmadre, Jack III-467  
 Van Gool, Luc I-143, I-369, I-452,  
 I-620, I-734, III-425  
 Vasquez, Dizan V-296  
 Vasudevan, Ram III-101  
 Vazquez-Reina, Amelio V-268  
 Veeraraghavan, Ashok I-100, II-237  
 Veksler, Olga V-211  
 Verbeek, Jakob I-634  
 Vese, Luminita I-185  
 Vicente, Sara II-465  
 Villanueva, Juan J. VI-280  
 Vondrick, Carl IV-610  
 von Lavante, Etienne V-743  
 Vu, Ngoc-Son I-313
- Wah, Catherine IV-438  
 Walk, Stefan VI-182  
 Wang, Bo III-328, V-15  
 Wang, Chen I-257  
 Wang, Gang V-169  
 Wang, Hua III-762, IV-793, VI-126  
 Wang, Huayan II-435, IV-497  
 Wang, Jue V-617  
 Wang, Kai I-591  
 Wang, Lei III-524  
 Wang, Liang I-257, IV-708  
 Wang, Peng II-608  
 Wang, Qifan IV-525  
 Wang, Shengnan IV-87  
 Wang, Xiaogang VI-420  
 Wang, Xiaosong V-478  
 Wang, Xiaoyu III-200  
 Wang, Xinggang III-328, V-15  
 Wang, Yang II-580, V-155  
 Wang, Zengfu V-729  
 Wang, Zhengxiang I-706  
 Watanabe, Takuya VI-337  
 Wedel, Andreas IV-582  
 Weickert, Joachim IV-568  
 Weinland, Daniel III-635  
 Weiss, Yair I-762  
 Welinder, Peter IV-438  
 Werman, Michael II-749  
 Wheeler, Frederick W. I-354  
 Wilburn, Bennett I-59  
 Wildes, Richard I-563, IV-511

- Wojek, Christian IV-467  
 Wong, Tien-Tsin V-422  
 Wu, Changchang II-142, IV-368  
 Wu, Jianxin II-552  
 Wu, Szu-Wei I-271  
 Wu, Xiaolin VI-351  
 Wu, Ying I-411, I-480  
 Wyatt, Jeremy L. IV-58
- Xavier, João IV-283  
 Xia, Yan V-729  
 Xiao, Jianxiong V-561  
 Xie, Xianghua IV-680  
 Xing, Eric P. V-85, V-785  
 Xu, Bing-Xin V-658  
 Xu, Li I-157  
 Xu, Wei VI-434
- Yamato, Junji III-215  
 Yan, Junchi III-790  
 Yan, Shuicheng III-748  
 Yang, Jianchao III-566, V-113  
 Yang, Jie III-790  
 Yang, Lin IV-624  
 Yang, Meng VI-448  
 Yang, Qingxiong IV-87  
 Yang, Ruigang IV-708  
 Yang, Xingwei III-411, V-450, V-757  
 Yang, Yezhou V-631  
 Yao, Angela III-425  
 Yarlagadda, Pradeep V-197  
 Yau, Shing-Tung V-672  
 Yeh, Tom II-693  
 Yezzi, Anthony I-73, VI-71  
 Yilmaz, Alper IV-410  
 Young, Peter IV-15  
 Yu, Chanki IV-115  
 Yu, Jin V-533  
 Yu, Jingyi I-15  
 Yu, Kai VI-434, V-113, V-141  
 Yu, Xiaodong V-127
- Yuan, Jing VI-379  
 Yuan, Xiaoru I-257  
 Yuen, Jenny II-707  
 Yuille, Alan IV-539, V-43
- Zach, Christopher IV-354  
 Zaharescu, Andrei I-563  
 Zeng, Wei V-672  
 Zeng, Zhi VI-532  
 Zhang, Cha III-229  
 Zhang, Chenxi IV-708  
 Zhang, Guofeng V-422  
 Zhang, Haichao III-566  
 Zhang, Honghui V-561  
 Zhang, Junping I-257  
 Zhang, Lei IV-157, VI-448, V-506  
 Zhang, Shaoting III-607  
 Zhang, Tong V-141  
 Zhang, Wei I-115, VI-420  
 Zhang, Yanning III-566  
 Zhang, Yuhang III-524  
 Zhang, Zhengyou III-229  
 Zhao, Bin V-785  
 Zhao, Mingtian IV-101  
 Zhao, Qiping IV-101  
 Zhao, Yong VI-351  
 Zheng, Ke Colin III-621  
 Zheng, Wenming VI-490  
 Zheng, Ying III-663  
 Zhou, Changyin VI-337  
 Zhou, Jun II-636  
 Zhou, Qian-Yi III-115  
 Zhou, Xi V-141  
 Zhou, Yue III-790  
 Zhou, Zhenglong II-265  
 Zhu, Long (Leo) V-43  
 Zhu, Song-Chun IV-101  
 Zickler, Todd I-45, II-251  
 Zimmer, Henning IV-568  
 Zisserman, Andrew III-677  
 Zitnick, C. Lawrence II-170