

Frank Dellaert Jan-Michael Frahm  
Marc Pollefeys Laura Leal-Taixé  
Bodo Rosenhahn (Eds.)

LNCS 7474

# Outdoor and Large-Scale Real-World Scene Analysis

15th International Workshop on  
Theoretical Foundations of Computer Vision  
Dagstuhl Castle, Germany, June/July 2011  
Revised Selected Papers



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*

Frank Dellaert Jan-Michael Frahm  
Marc Pollefeys Laura Leal-Taixé  
Bodo Rosenhahn (Eds.)

# Outdoor and Large-Scale Real-World Scene Analysis

15th International Workshop  
on Theoretical Foundations of Computer Vision  
Dagstuhl Castle, Germany, June 26 - July 1, 2011  
Revised Selected Papers

Volume Editors

Frank Dellaert

Georgia Institute of Technology, College of Computing Building  
Atlanta, GA 30332-0280, USA  
E-mail: frank@cc.gatech.edu

Jan-Michael Frahm

University of North Carolina at Chapel Hill, Department of Computer Science  
Chapel Hill, NC 27599, USA  
E-mail: jmf@cs.unc.edu

Marc Pollefeys

ETH Zurich, CVG - Institute of Visual Computing  
8092 Zurich, Switzerland  
E-mail: marc.pollefeys@inf.ethz.ch

Laura Leal-Taixé

Bodo Rosenhahn  
Leibniz Universität, Institute for Information Processing (TNT)  
30167 Hannover, Germany  
E-mail: rosenhahn@tnt.uni-hannover.de

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-34090-1

e-ISBN 978-3-642-34091-8

DOI 10.1007/978-3-642-34091-8

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012948536

CR Subject Classification (1998): I.4.8, I.2.10, I.4, F.2.2, I.5.3-4, J.2, I.2.6

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

© Springer-Verlag Berlin Heidelberg 2012

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.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

The topic of the 15th Workshop on Theoretical Foundations of Computer Vision was *Outdoor and Large-Scale Real-World Scene Analysis*, which covers all aspects, applications, and open problems regarding the performance or design of computer vision algorithms capable of working in outdoor set-ups and/or large-scale environments. Developing these methods is important for driver assistance, city modeling and reconstruction, virtual tourism, telepresence, and motion capture. With this workshop we aimed to attain several objectives, outlined below.

The first objective of the workshop was to take stock of the performance of existing state-of-the-art computer vision algorithms and to define metrics and benchmark datasets on which to evaluate them. It is imperative that we push existing algorithms, which are currently benchmarked or tested with artificial or indoor set-ups, toward *real* applications. Methods of interest are 3D reconstruction, optic flow computation, motion capture, surveillance, object recognition, and tracking. These need to be dragged out of the lab and into the real world. Over the last few years the computer vision community has recognized this problem and several groups are increasingly concentrating on the analysis of uncontrolled scenes. Examples include reconstructing large city models from online image collections such as Flickr, or human tracking and behavior recognition in TV footage or video from arbitrary outdoor scenes. An outcome we envision is the definition of appropriate metrics, benchmark sequences, and the definition of a *grand-challenge problem* that exposes algorithms to all the difficulties associated with large-scale outdoor scenes while simultaneously mobilizing the research community.

The second objective, then, was to define what the open problems are and which aspects of outdoor and large-scale scene analysis make the problem currently intractable. In uncontrolled, outdoor settings many problems start to arise, among them harsh viewing conditions, changing lighting conditions, and artifacts from wind, rain, clouds, or temperature etc. In addition, large-scale modeling, i.e., spanning city-scale areas, contains difficult challenges of data association and self-consistency that simply do not appear in smaller data-sets. Failure of basic building-block algorithms seems likely or even inevitable, requiring system-level approaches in order to be robust to failure. One of the difficulties lies in the fact that the observer loses complete control over the scene, which can become arbitrary complex. This also brings with it the challenge of describing the scene in terms other than purely geometric, i.e., perform true scene *understanding* at multiple spatial and temporal scales. Finally, outdoor scenes are dynamic and changing over time, requiring event learning and understanding as well as integrating behavior recognition. In this regard, we brought in participants from industry in order to ground the challenges discussed in real-world, useful applications.

The third and final objective was to discuss strategies that address these challenges, by bringing together a diverse set of international researchers with people interested in the applications, e.g., arising from photogrammetry, geoinformatics, driver-assistance systems, or human motion analysis. Although these people work in different fields and communities, they are unified by their goal of dealing with images and/or video from outdoor scenes and uncontrolled settings. In the workshop we allowed for an exchange of different modeling techniques and experiences researchers have collected. We allowed time for working groups during the workshop that connect people and whose goals are to develop ideas/roadmaps; additionally, we allowed young researchers to connect with senior researchers, and in general allowed for an exchange between researchers who would usually not meet otherwise.

We are grateful to the team at Castle Dagstuhl for supporting our workshop. We would like to thank all participants for their encouraging presentations, lively discussions, and contributions for this book. The published papers were carefully selected after a blind per-review process and reflect major topics presented at the seminar.

June 2012

Frank Dellaert  
Jan-Michael Frahm  
Marc Pollefeys  
Laura Leal-Taixé  
Bodo Rosenhahn

# Organization

The 15th Workshop Theoretic Foundations of Computer Vision, titled “Outdoor and Large-Scale Real-World Scene Analysis,” was organized by Frank Dellaert, Jan-Michael Frahm, Marc Pollefeys, and Bodo Rosenhahn.

## Executive Committee

### Organizers

Frank Dellaert	Georgia Institute of Technology, USA
Jan-Michael Frahm	University of North Carolina, Chapel Hill, USA
Marc Pollefeys	ETH Zürich, Switzerland
Bodo Rosenhahn	Leibniz Universität Hannover, Germany

### Edited in Cooperation with

Laura Leal-Taixé	Leibniz Universität Hannover, Germany
------------------	---------------------------------------

## Participants

Steffen Abraham	Robert Bosch GmbH - Hildesheim, Germany
Sameer Agarwal	Google - Seattle, USA
Ioannis Brilakis	Georgia Institute of Technology, USA
Gabriel Brostow	University College London, UK
Andrés Bruhn	Universität des Saarlandes, Germany
Daniel Cremers	TU München, Germany
Frank Dellaert	Georgia Institute of Technology, USA
Ralf Dragon	Leibniz Universität Hannover, Germany
Wolfgang Förstner	Universität Bonn, Germany
Jan-Michael Frahm	University of North Carolina, Chapel Hill, USA
Jean-Sebastien Franco	INRIA Rhône-Alpes, France
Friedrich Fraundorfer	ETH Zürich, Switzerland
Jürgen Gall	ETH Zürich, Switzerland
Stefan Gehrig	Daimler Research - Stuttgart, Germany
Michael Goesele	TU Darmstadt, Germany
Radek Grzeszczuk	NRC - Palo Alto, USA
Johan Hedborg	Linköping University, Sweden
Christian Heipke	Leibniz Universität Hannover, Germany
Thomas Helten	MPI für Informatik - Saarbrücken, Germany

## VIII Organization

Vaclav Hlavac	Czech Technical University, Czech Republic
Atsushi Imiya	Chiba University, Japan
Gisela Klette	Auckland University of Technology, New Zealand
Reinhard Klette	University of Auckland, New Zealand
Felix Klose	TU Braunschweig, Germany
Reinhard Koch	Universität Kiel, Germany
Daniel Kondermann	Universität Heidelberg, Germany
Norbert Krüger	University of Southern Denmark - Odense, Denmark
Laura Leal-Taixé	Leibniz Universität Hannover, Germany
Kshitij Marwah	MIT - Cambridge, USA
Helmut Mayer	Universität der Bundeswehr - München, Germany
Bärbel Mertsching	Universität Paderborn, Germany
Rudolf Mester	Universität Frankfurt, Germany
Meinard Müller	MPI für Informatik - Saarbrücken, Germany
Thomas Pajdla	Czech Technical University, Czech Republic
Marc Pollefeys	ETH Zürich, Switzerland
Gerard Pons-Moll	Leibniz Universität Hannover, Germany
Dan Raviv	Technion - Haifa, Israel
Ralf Reulke	HU Berlin, Germany
Bodo Rosenhahn	Leibniz Universität Hannover, Germany
Torsten Sattler	RWTH Aachen, Germany
Silvio Savarese	University of Michigan, USA
Andreas Schilling	Universität Tübingen, Germany
Falko Schindler	Universität Bonn, Germany
Thorsten Thormählen	MPI für Informatik - Saarbrücken, Germany
Tinne Tuytelaars	K.U. Leuven, Belgium
Michael Wand	MPI für Informatik - Saarbrücken, Germany
Jan Dirk Wegner	Leibniz Universität Hannover, Germany
Christopher M. Zach	ETH Zürich, Switzerland
Henning Zimmer	Universität des Saarlandes, Germany



# Table of Contents

Exploiting Pedestrian Interaction via Global Optimization and Social Behaviors . . . . .	1
<i>Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn</i>	
An Evaluation Framework for Stereo-Based Driver Assistance . . . . .	27
<i>Nicolai Schneider, Stefan Gehrig, David Pfeiffer, and Konstantinos Banitsas</i>	
Real-World Stereo-Analysis Evaluation . . . . .	52
<i>Sandino Morales, Simon Hermann, and Reinhard Klette</i>	
Pyramid Transform and Scale-Space Analysis in Image Analysis . . . . .	78
<i>Yoshihiko Mochizuki and Atsushi Imiya</i>	
Towards Feature-Based Situation Assessment for Airport Apron Video Surveillance . . . . .	110
<i>Ralf Dragon, Michele Fenzi, Wolf Siberski, Bodo Rosenhahn, and Jörn Ostermann</i>	
Generalized Subgraph Preconditioners for Large-Scale Bundle Adjustment . . . . .	131
<i>Yong-Dian Jian, Doru C. Balcan, and Frank Dellaert</i>	
Achievements and Challenges in Recognizing and Reconstructing Civil Infrastructure . . . . .	151
<i>Ioannis Brilakis, Fei Dai, and Stefania-Christina Radopoulou</i>	
Equi-affine Invariant Geometries of Articulated Objects . . . . .	177
<i>Dan Raviv, Alexander M. Bronstein, Michael M. Bronstein, Ron Kimmel, and Nir Sochen</i>	
Towards Fast Image-Based Localization on a City-Scale . . . . .	191
<i>Torsten Sattler, Bastian Leibe, and Leif Kobbelt</i>	
Perspective and Non-perspective Camera Models in Underwater Imaging – Overview and Error Analysis . . . . .	212
<i>Anne Sedlazeck and Reinhard Koch</i>	
An Introduction to Random Forests for Multi-class Object Detection . . .	243
<i>Juergen Gall, Nima Razavi, and Luc Van Gool</i>	
Segmentation and Classification of Objects with Implicit Scene Context . . . . .	264
<i>Jan D. Wegner, Bodo Rosenhahn, and Uwe Sörgel</i>	

Dense 3D Reconstruction from Wide Baseline Image Sets . . . . .	285
<i>Helmut Mayer, Jan Bartelsen, Heiko Hirschmüller, and Andreas Kuhn</i>	
Data-Driven Manifolds for Outdoor Motion Capture . . . . .	305
<i>Gerard Pons-Moll, Laura Leal-Taixé, Juergen Gall, and Bodo Rosenhahn</i>	
On Performance Analysis of Optical Flow Algorithms . . . . .	329
<i>Daniel Kondermann, Steffen Abraham, Gabriel Brostow, Wolfgang Förstner, Stefan Gehrig, Atsushi Imiya, Bernd Jähne, Felix Kloese, Marcus Magnor, Helmut Mayer, Rudolf Mester, Tomas Pajdla, Ralf Reulke, and Henning Zimmer</i>	
Camera-Based Fall Detection on Real World Data . . . . .	356
<i>Glen Debard, Peter Karsmakers, Mieke Deschodt, Ellen Vlaeyen, Eddy Dejaeger, Koen Milisen, Toon Goedemé, Bart Vanrumste, and Tinne Tuytelaars</i>	
Semantic Structure from Motion: A Novel Framework for Joint Object Recognition and 3D Reconstruction . . . . .	376
<i>Sid Yingze Bao and Silvio Savarese</i>	
Hierarchical Surface Reconstruction from Multi-resolution Point Samples . . . . .	398
<i>Ronny Klowsky, Patrick Mücke, and Michael Goesele</i>	
Traffic Observation and Situation Assessment . . . . .	419
<i>Ralf Reulke, Dominik Rueß, Kristian Manthey, and Andreas Luber</i>	
<b>Author Index . . . . .</b>	<b>443</b>

# Exploiting Pedestrian Interaction via Global Optimization and Social Behaviors

Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn

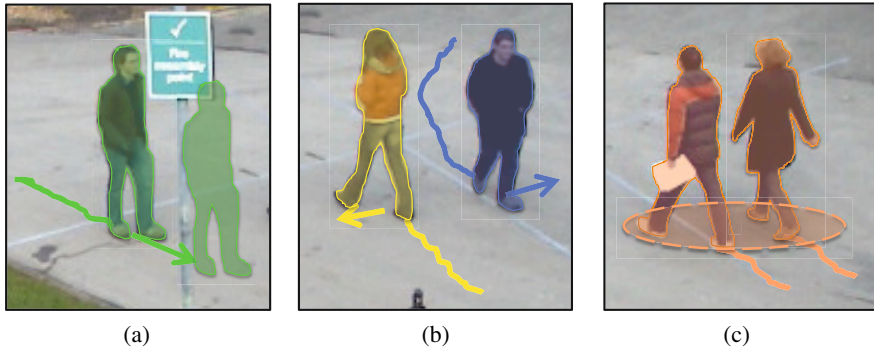
Leibniz Universität Hannover, Appelstr. 9A, Hannover, Germany  
{leal, pons, rosenhahn}@tnt.uni-hannover.de

**Abstract.** Multiple people tracking consists in detecting the subjects at each frame and matching these detections to obtain full trajectories. In semi-crowded environments, pedestrians often occlude each other, making tracking a challenging task. Tracking methods mostly work with the assumption that each pedestrian moves independently unaware of the objects or the other pedestrians around it. In the real world though, it is clear that when walking in a crowd, pedestrians try to avoid collisions, keep a close distance to a group of friends or avoid static obstacles in the scene.

In this paper, we present an approach which includes the interaction between pedestrians in two ways: first, including social and grouping behavior as a physical model within the tracking system, and second, using a global optimization scheme which takes into account all trajectories and all frames to solve the data association problem. Results are presented on three challenging publicly available datasets, showing our method outperforms state-of-the-art tracking systems. We also make a thorough analysis of the effect of the parameters of the proposed tracker as well as its robustness against noise, outliers and missing data.

## 1 Introduction

Multiple people tracking is a key problem for many computer vision tasks, such as surveillance, animation or activity recognition. In crowded environments occlusions and false detections are common, and although there have been substantial advances in the last years, tracking is still a challenging task. Tracking is often divided in two steps: detection, finding the objects of interest on every frame, and data association, matching the detections to form complete trajectories in time. Researchers have presented improvements on the object detector [1-3] as well as on the optimization techniques [4,5] and even specific algorithms have been developed for tracking in crowded scenes [6,7]. Though each object can be tracked separately, recent works have proven that tracking objects jointly and taking into consideration their interaction can give much better results in complex scenes. Current research is mainly focused on two aspects to exploit the interaction between pedestrians: the use of a global optimization strategy [8,9] and a social motion model [10,11]. The focus of this paper is to marry the concepts of global optimization and social and grouping behavior to obtain a robust tracker able to work in crowded scenarios. We extend the work presented in [12] to include more theoretical details, experimental results and details about the performance of the proposed method.



**Fig. 1.** Including social and grouping behavior to the network flow graph. (a) Constant velocity assumption. (b) Avoidance forces. (c) Group attraction forces.

## 1.1 Related Work

The optimization strategy deals with the data association problem, which is usually solved on a frame-by-frame basis or one track at a time. Several methods can be used such as Markov Chain Monte Carlo (MCMC) [13], multi-level Hungarian [14], inference in Bayesian networks [15] or the Nash Equilibrium of game theory [16]. In [17] an efficient approximative Dynamic Programming (DP) scheme is presented, in which trajectories are estimated one after the other. This means that if a trajectory is formed using a certain detection, the other trajectories which are computed later will not be able to use that detection anymore. This obviously does not guarantee a global optimum for all trajectories. Recent works show that global optimization can be more reliable in crowded scenes as it solves the matching problem jointly for all tracks. The multiple object tracking problem is defined as a linear constrained optimization flow problem and Linear Programming (LP) is commonly used to find the global optimum. The idea was first used for people tracking in [18], although this method needs to know a priori the number of targets to track, which limits its application in real tracking situations. In [9], the scene is divided into identical cells, each represented by a node in the constructed graph. Using the information of the Probability Occupancy Map, the problem is formulated either as a max-flow and solved with Simplex, or as a min-cost and solved using k-shortest paths, which is a more efficient solution. Both methods show a far superior performance when compared to the same approach with DP [17]. The authors of [19] also define the problem as a maximum flow on an hexagonal grid, but instead of using matching individual detections, they make use of tracklets. This has the advantage that they can precompute the social forces for each of these tracklets, nonetheless, the fact that the tracklets are chosen locally, means the overall matching is not truly global, and if errors occur during the creation of the tracklets, these cannot be overcome by the global optimization. In [20], global and local methods are combined to match trajectories across cameras and across time. Finally, in [8] the tracking problem is formulated as a Maximum A-Posteriori (MAP) problem, which is mapped to a minimum-cost network flow and then efficiently solved using LP. In this case, each node represents a detection, which means the graph is much smaller compared to [9, 19].

Most tracking systems work with the assumption that the motion model for each target is independent. This simplifying assumption is especially problematic in crowded scenes: imagine the chaos if every pedestrian followed his or her chosen path and completely ignored the other pedestrians in the scene. In order to avoid collisions and reach the chosen destination at the same time, a pedestrian follows a series of social rules or social forces. These have been defined in what is called the Social Force Model (SFM) [21], which has been used for abnormal crowd behavior detection [22], crowd simulation [23] and has only recently been applied to multiple people tracking: in [24], an energy minimization approach is used to predict the future position of each pedestrian considering all the terms of the social force model. In [10] and [25], the social forces are included in the motion model of the Kalman or Extended Kalman filter. In [26] a method is presented to detect small groups of people in a crowd, but it is only recently that grouping behavior has been included in a tracking framework [11,27,28]. In [28] groups are included in a graphical model which contains cycles and, therefore, Dual Decomposition [29] is needed to find the solution, which obviously is computationally much more expensive than using Linear Programming. Moreover, the results presented in [28] are only for short time windows. On the other hand, the formulations of [11,27] are predictive by nature and therefore too local and unable to deal with trajectory changes (e.g. when people meet and stop to talk).

Social behavior models have only been introduced within a predictive framework, which are suboptimal due to the recursive nature of filtering. Therefore, in contrast to previous works, we propose to include social and grouping models into a global optimization framework which allows us to better estimate the true maximum a-posteriori probability of the trajectories.

## 1.2 Contributions

We present a novel approach for multiple people tracking which takes into account the interaction between pedestrians in two ways: first, using global optimization for data association and second, including social as well as grouping behavior. The key insight is that people plan their trajectories in advance in order to avoid collisions, therefore, a graph model which takes into account future and past frames is the perfect framework to include social and grouping behavior. We formulate multiple object tracking as a minimum-cost network flow problem, and present a new graph model which yields to better results than existing global optimization approaches. The social force model (SFM) and grouping behavior (GR) are included in an efficient way without altering the linearity of the problem. Results on several challenging public datasets show the improvement of the tracking results in crowded environments. Experiments with missing data, noise and outliers are also shown to test the robustness of the proposed approach. In this paper, we extend the work presented in [12] in three aspects : (i) more detailed theoretical explanations and background on Linear Programming for multiple object tracking; (ii) experimental results with different parameter values to see the effect of each of them on tracking results and (iii) detailed implementation details and computational aspects of the proposed method.

## 2 Multiple People Tracking

Tracking is commonly divided in two steps: object detection and data association. First, the objects are detected in each frame of the sequence and second, the detections are matched to form complete trajectories. In this section we define the data association problem and describe how to convert it to a minimum-cost network flow problem, which can be efficiently solved using Linear Programming.

The idea is to build a graph in which the nodes represent the pedestrian detections. These nodes are fully connected to past and future observations by edges, which determine the relation between two observations with a cost. Thereby, the matching problem is equivalent to a minimum-cost network flow problem: finding the optimal set of trajectories is equivalent to sending flow through the graph so as to minimize the cost. This can be efficiently computed using the Simplex algorithm or k-shortest paths [30].

### 2.1 Problem Statement

Let  $\mathcal{O} = \{\mathbf{o}_k t\}$  be a set of object detections with  $\mathbf{o}_k^t = (\mathbf{p}_k, t)$ , where  $\mathbf{p}_k = (x, y, z)$  is the 3D position and  $t$  is the time stamp. A trajectory is defined as a list of ordered object detections  $T_k = \{\mathbf{o}_k^1, \mathbf{o}_k^2, \dots, \mathbf{o}_k^N\}$ , and the goal of multiple object tracking is to find the set of trajectories  $\mathcal{T}^* = \{T_k\}$  that best explains the detections. This is equivalent to maximizing the a-posteriori probability of  $\mathcal{T}$  given the set of detections  $\mathcal{O}$ . Assuming detections are conditionally independent, the objective function is expressed as:

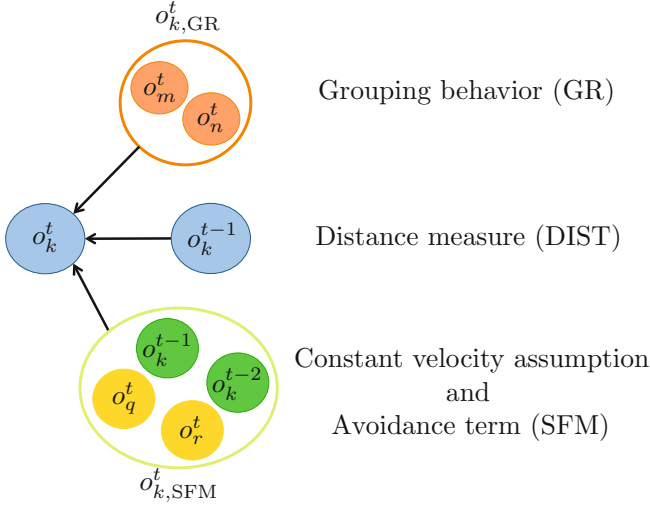
$$\mathcal{T}^* = \underset{\mathcal{T}}{\operatorname{argmax}} P(\mathcal{T}|\mathcal{O}) = \underset{\mathcal{T}}{\operatorname{argmax}} \prod_k P(\mathbf{o}_k|\mathcal{T})P(\mathcal{T}) \quad (1)$$

$P(\mathbf{o}_k|\mathcal{T})$  is the likelihood of the detection. In order to reduce the space of  $\mathcal{T}$ , we make the assumption that the trajectories cannot overlap (i.e., a detection cannot belong to two trajectories), but unlike [8], we do not define the motion of each subject to be independent, therefore, we deal with a much larger search space. We extend this space by including the following dependencies for each trajectory  $T_k$ :

- Constant velocity assumption: the observation  $\mathbf{o}_k^t \in T_k$  depends on past observations  $[\mathbf{o}_k^{t-1}, \mathbf{o}_k^{t-2}]$
- Grouping behavior: If  $T_k$  belongs to a group, the set of members of the group  $\mathcal{T}_{k,\text{GR}}$  has an influence on  $T_k$
- Avoidance term:  $T_k$  is affected by the set of trajectories  $\mathcal{T}_{k,\text{SFM}}$  which are close to  $T_k$  at some point in time and do not belong to the same group as  $T_k$

The first and third dependencies are grouped into the SFM term. The sets  $\mathcal{T}_{k,\text{SFM}}$  and  $\mathcal{T}_{k,\text{GR}}$  are disjoint, i.e., a pedestrian can have an attractive effect or a repulsive effect on another pedestrian, but not both. Therefore, we can assume that these two terms are independent and decompose  $P(\mathcal{T})$  as:

$$\begin{aligned} P(\mathcal{T}) &= \prod_{T_k \in \mathcal{T}} P(T_k \cap \mathcal{T}_{k,\text{SFM}} \cap \mathcal{T}_{k,\text{GR}}) \\ &= \prod_{T_k \in \mathcal{T}} P(\mathcal{T}_{k,\text{SFM}}|T_k)P(\mathcal{T}_{k,\text{GR}}|T_k)P(T_k) \end{aligned} \quad (2)$$



**Fig. 2.** Diagram of the dependencies for each observation  $\mathbf{o}_k^t$

where the trajectories are represented by a Markov chain:

$$\begin{aligned}
 P(\mathcal{T}) = \prod_{T_k \in \mathcal{T}} P_{\text{in}}(\mathbf{o}_k^1) \dots P(\mathbf{o}_k^t | \mathbf{o}_k^{t-1}) \\
 P_{k,\text{SFM}}(\mathbf{o}_k^t | \mathbf{o}_{k,\text{SFM}}^t, \mathbf{o}_k^{t-1}) P_{k,\text{GR}}(\mathbf{o}_k^t | \mathbf{o}_{k,\text{GR}}^t, \mathbf{o}_k^{t-1}) \\
 \dots P_{\text{out}}(\mathbf{o}_k^N)
 \end{aligned} \quad (3)$$

where  $P_{\text{in}}(\mathbf{o}_k^t)$  is the probability that a trajectory is initiated with detection  $\mathbf{o}_k^t$ ,  $P_{\text{out}}(\mathbf{o}_k^t)$  the probability that the trajectory is terminated at  $\mathbf{o}_k^t$  and  $P(\mathbf{o}_k^t | \mathbf{o}_k^{t-1})$  is the probability that  $\mathbf{o}_k^{t-1}$  is followed by  $\mathbf{o}_k^t$  in the trajectory.  $P_{k,\text{SFM}}$  evaluates how well the social rules are kept if  $\mathbf{o}_k^t$  is matched to  $\mathbf{o}_k^{t-1}$ , and  $P_{k,\text{GR}}$  describes how well the structure of the group is kept.

Let us assume that we are analyzing observation  $\mathbf{o}_k^t$ . In Figure 2 we summarize which observations influence the matching of  $\mathbf{o}_k^t$ . Typical approaches [8] only take into account distance (DIST) information, that is, the observation in the previous frame  $\mathbf{o}_k^{t-1}$ . We introduce the social dependencies (SFM) given by the constant velocity assumption (green nodes) and the avoidance term (yellow nodes). In this case, two observations,  $\mathbf{o}_q^t$  and  $\mathbf{o}_r^t$  that do not belong to the same group as  $\mathbf{o}_k^t$ , will be considered to create a repulsion effect on  $\mathbf{o}_k^t$ . On the other hand, the orange nodes which depict the grouping term (GR), are two other observations  $\mathbf{o}_m^t$  and  $\mathbf{o}_n^t$  which do belong to the same group as  $\mathbf{o}_k^t$  and therefore have an attraction effect on  $\mathbf{o}_k^t$ . Note that all these dependencies can only be modeled by high order terms, which means that either we use complex solvers [28] to find a solution in graphs with cycles, or we keep the linearity of the problem by using an iterative approach as we explain later on.

## 2.2 Tracking with Linear Programming

We linearize the objective function by defining a set of flow flags  $f_{i,j} = \{0, 1\}$  which indicate if an edge  $(i, j)$  is in the path of a trajectory or not. In a minimum cost network flow problem, the objective is to find the values of the variables that minimize the total cost of the flows over the network. Defining the costs as negative log-likelihoods, and combining Equations (1), (2) and (3), the following objective function is obtained:

$$\begin{aligned}
 \mathcal{T}^* &= \underset{\mathcal{T}}{\operatorname{argmin}} \sum_{T_k \in \mathcal{T}} -\log P(T_k) - \log P(\mathcal{T}_{\text{SFM}}|T_k) \\
 &\quad - \log P(\mathcal{T}_{\text{GR}}|T_k) + \sum_k -\log P(\mathbf{o}_k|\mathcal{T}) \\
 &= \underset{\mathcal{T}}{\operatorname{argmin}} \sum_i C_{\text{in},i} f_{\text{in},i} + \sum_i C_{i,\text{out}} f_{i,\text{out}} \\
 &\quad + \sum_{i,j} (C_{i,j} + C_{\text{SFM},i,j} + C_{\text{GR},i,j}) f_{i,j} + \sum_i C_i f_i
 \end{aligned} \tag{4}$$

subject to the following constraints:

- Edge capacities: we assume that each detection can only correspond to one trajectory, therefore, the edge capacities have an upper bound of  $u_{ij} \leq 1$  and:

$$f_{\text{in},i} + f_i \leq 1 \quad f_{i,\text{out}} + f_i \leq 1 \tag{5}$$

- Flow conservation at the nodes:

$$f_{\text{in},i} + f_i = \sum_j f_{i,j} \quad \sum_j f_{j,i} = f_{i,\text{out}} + f_i \tag{6}$$

- Exclusion property:

$$f_{i,j} = \{0, 1\} \tag{7}$$

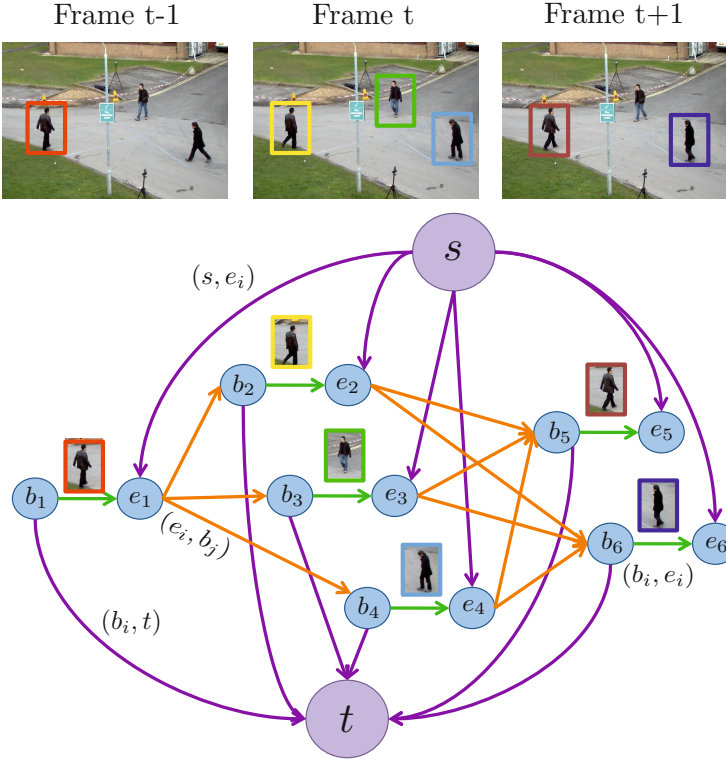
The condition in Eq. 7 requires us to solve an integer program, which is known to be NP-complete. Nonetheless, we can relax the condition to have the following linear equation:

$$0 \leq f_{i,j} \leq 1. \tag{8}$$

Now the problem is defined and can be solved as a linear program. If certain conditions are fulfilled, the solution  $\mathcal{T}^*$  will still be integer, and therefore will also be the optimal solution to the initial integer program. We discuss the integrality of the solution in more detail in Section 4.

To map this formulation into a cost-flow network, we define  $G = (N, E)$  to be a directed network with a cost  $C_{i,j}$  and a capacity  $u_{ij}$  associated with every edge  $(i, j) \in E$ . An example of such a network is shown in Figure 3; it contains two special nodes, the source  $s$  and the sink  $t$ ; all flow that goes through the graph starts at the  $s$  node and ends at the  $t$  node. Thereby, each flow represents a trajectory  $T_k$  and the path that





**Fig. 3.** Example of a graph with the special source  $s$  and sink  $t$  nodes, 6 detections which are represented by two nodes each: the beginning  $b_i$  and the end  $e_i$

each flow follows indicates which observations belong to each of the trajectories. Each observation  $\mathbf{o}_i$  is represented with two nodes, the beginning node  $b_i \in N$  and the end node  $e_i \in N$  (see Figure 3). A detection edge connects  $b_i$  and  $e_i$ .

Below we detail the three types of edges present in the graphical model and the cost for each type:

**Link Edges.** The edges  $(e_i, b_j)$  connect the end nodes  $e_i$  with the beginning nodes  $b_j$  in following frames, with cost  $C_{i,j}$  and flow  $f_{i,j}$ , defined as:

$$f_{i,j} = \begin{cases} 1, & \mathbf{o}_i \text{ and } \mathbf{o}_j \text{ belong to } T_k \text{ and } \Delta f \leq F_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

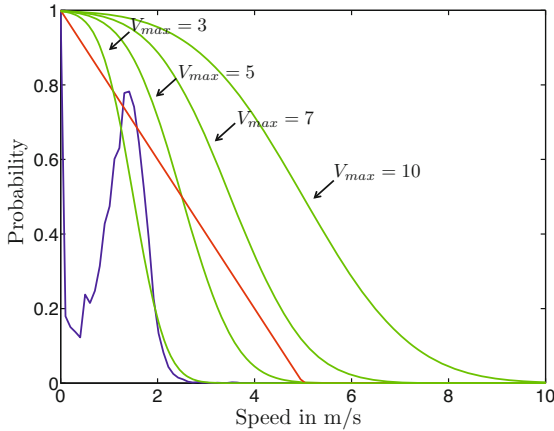
where  $\Delta f$  is the frame number difference between nodes  $j$  and  $i$  and  $F_{\max}$  is the maximum allowed frame gap.

The costs of the link edges represent the spatial relation between different subjects. Assuming that a subject cannot move a lot from one frame to the next, we define the costs to be a decreasing function of the distance between detections in successive

frames. The time gap between observations is also taken into account in order to be able to work at any frame rate, therefore velocity measures are used instead of distances. The velocities are mapped to probabilities with a Gauss error function as shown in Equation (10), assuming the pedestrians cannot exceed a maximum velocity  $V_{\max}$ . The effect of parameter  $V_{\max}$  is detailed in Section 5.1

$$E(V_t, V_{\max}) = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left( \frac{-V_t + \frac{V_{\max}}{2}}{\frac{V_{\max}}{4}} \right) \quad (10)$$

As we can see in Figure 4, the advantage of using Equation (10) over a linear function is that the probability of lower velocities decreases more slowly, while the probability for higher velocities decreases more rapidly. This is consistent with the probability distribution of speed learned from training data.



**Fig. 4.** Blue = normalized histogram of speeds learned from training data. Red = probability distribution if cost depends linearly on the velocity. Green = probability distribution if the relation of cost and velocities is expressed by Equation (10). An  $V_{\max} = 7m/s$  is used in the experiments.

Therefore, the cost of a link edge is defined as:

$$\begin{aligned} C_{i,j} &= -\log(P(\mathbf{o}_j|\mathbf{o}_i)) + C(\Delta f) \\ &= -\log E \left( \frac{\|\mathbf{p}_i - \mathbf{p}_i\|}{\Delta t}, V_{\max} \right) + C(\Delta f) \end{aligned} \quad (11)$$

where  $C(\Delta f) = -\log(B_j^{\Delta f-1})$  is the cost depending on the frame difference between detections.

**Detection Edges.** The edges  $(b_i, e_i)$  connect the beginning node  $b_i$  and end node  $e_i$ , with cost  $C_i$  and flow  $f_i$ , defined as:

$$f_i = \begin{cases} 1, & \mathbf{o}_i \text{ belongs to } T_k \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

If all the costs of the edges are positive, the solution to the minimum-cost problem is the trivial null flow. Consequently, we represent each observation with two nodes and a detection edge with negative cost:

$$C_i = \log(1 - P_{det}(\mathbf{o}_i)) + \log\left(\frac{\text{BB}_{\min}}{\|\mathbf{p}_{\text{BB}} - \mathbf{p}_i\|}\right). \quad (13)$$

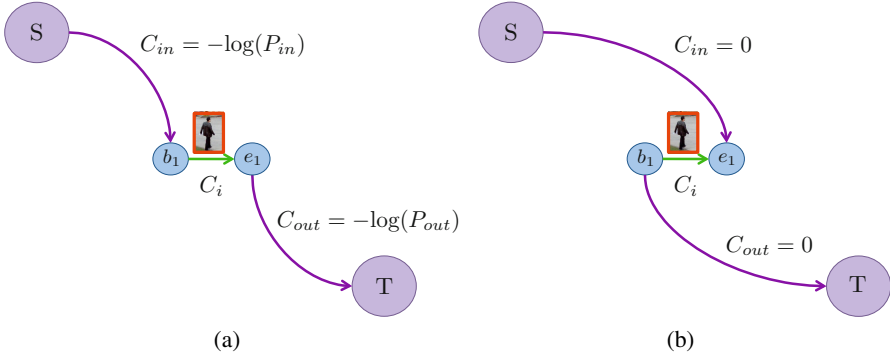
The higher the likelihood of a detection  $P_{det}(\mathbf{o}_i)$  the more negative the cost of the detection edge, hence, confident detections are likely to be in the path of the flow in order to minimize the total cost. If a map of the scene is available, we can also include this information in the detection cost. If a detection is far away from a possible entry/exit point, we add an extra negative cost to the detection edge, in order to favor that observation to be matched. The added cost depends on the distance to the closest entry/exit point  $\mathbf{p}_{\text{BB}}$ , and is only computed for distances higher than  $\text{BB}_{\min} = 1.5m$ . This is a probabilistic simple way of including other information present in the scene, such as obstacles or attraction points (shops, doors, etc).

**Entrance and Exit Edges.** The edges  $(s, e_i)$  connect the source  $s$  with all the end nodes  $e_i$ , with cost  $C_{\text{in},i}$  and flow  $f_{\text{in},i}$ . Similarly,  $(b_i, t)$  connects the end node  $b_i$  with sink  $t$ , with cost  $C_{i,\text{out}}$  and flow  $f_{i,\text{out}}$ . The flows are defined as:

$$f_{\text{in},i} \text{ (or } f_{i,\text{out}}) = \begin{cases} 1, & T_k \text{ starts (or ends) at } \mathbf{o}_i \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

In [8], the authors propose to create the opposite edges  $(s, b_i)$  and  $(e_i, t)$ , which means tracks entering and leaving the scene go through the detection node and therefore benefiting from its negative cost (see Figure 5(a)). If the costs  $C_{\text{in}}$  and  $C_{\text{out}}$  are then set to zero, a track will be started at each detection of each frame, because it will be cheaper to use the entrance and exit edges than the link edges. On the other hand, if  $C_{\text{in}}$  and  $C_{\text{out}}$  are very high, it will be hard for the graph to create any trajectory. Therefore, the choice of these two costs is extremely important. In [8], the costs are set according to the entrance and exit probabilities  $P_{\text{in}}$  and  $P_{\text{out}}$ , which are data dependent terms that need to be calculated during optimization.

In contrast, we propose to connect the  $s$  node with the end nodes and the  $t$  node to the begin nodes (as shown in Figure 5(b)). This way, we make sure that when a track starts (or ends) it does not benefit from the negative cost of the detection edge. Setting  $C_{\text{in}} = C_{\text{out}} = 0$  and taking into account the flow constraints of Eqs. (5) and (6), we make sure the trajectories are only created with the information of the link edges.



**Fig. 5.** (a) Graph structure as used in [8], which requires the computation of  $P_{in}$  and  $P_{out}$  in an Expectation-Maximization step during optimization. In contrast, the proposed graph structure in (b) allows us to get rid of these two extra parameters. The trajectories are found only with the information of the link and detection edges.

### 3 Modeling Social Behavior

If a pedestrian does not encounter any obstacles, the natural path to follow is a straight line. But what happens when the space gets more and more crowded and the pedestrian can no longer follow the straight path? Social interaction between pedestrians is especially important when the environment is crowded. In this section we consider how to include the social behavior [21], which we divide into the Social Force Model (SFM) and the Group behavior (GR), into our minimum-cost network flow problem.

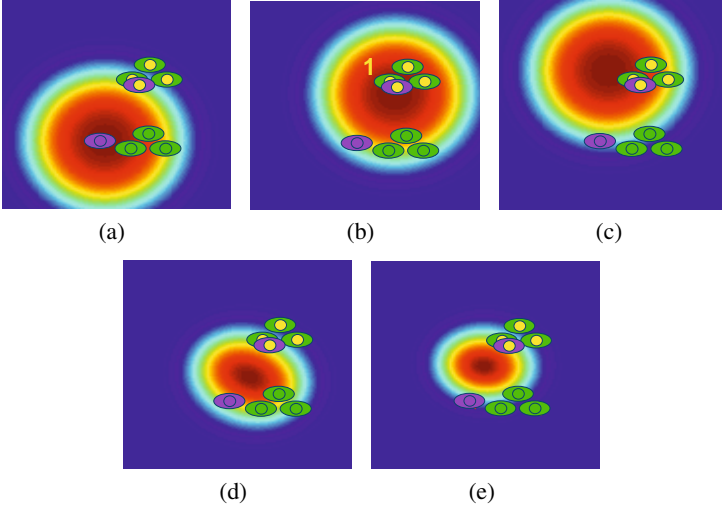
#### 3.1 Social Force Model

The social force model states that the motion of a pedestrian can be described as if they were subject to "social forces". There are three main terms that need to be considered: the desire of a pedestrian to maintain a certain speed, the desire to keep a comfortable distance from other pedestrians and the desire to reach a destination. Since we cannot know a priori the destination of the pedestrian in a real tracking system, we focus on the first two terms.

**Constant Velocity Assumption.** The pedestrian tries to keep a certain speed and direction, therefore we assume that in  $t + \Delta t$  we have the same speed as in  $t$  and predict the pedestrian's position in  $t + \Delta t$  accordingly.

$$\tilde{\mathbf{p}}_i^{t+\Delta t} = \mathbf{p}_i^t + \mathbf{v}_i^t \Delta t$$

**Avoidance Term.** The pedestrian also tries to avoid collisions and keep a comfortable distance from other pedestrians. We model this term as a repulsion field with an exponential distance-decay function with value  $\alpha$  learned from training data.



**Fig. 6.** Three green pedestrians walk in a group, the predicted positions in the next frame are marked by yellow heads. The purple pedestrian’s linearly predicted position (yellow head) clearly interferes with the trajectory of the group. Representation of the probability (blue is 0 red is 1) distribution for the purple’s next position using: [6\(a\)](#) only distances, [6\(b\)](#) only SFM (constant velocity assumption and avoidance term), [6\(c\)](#) only GR (considering the purple pedestrian belongs to the group), [6\(d\)](#) distances+SFM and [6\(e\)](#) distances+SFM+GR.

$$\mathbf{a}_i^{t+\Delta t} = \sum_{g_m \neq g_i} \exp \left( -\frac{\|\tilde{\mathbf{p}}_i^{t+\Delta t} - \tilde{\mathbf{p}}_m^{t+\Delta t}\|}{\alpha \Delta t} \right) \quad (15)$$

If we are computing the cost of edge  $(i, j)$ , we use the constant velocity assumption to predict the position of  $\mathbf{o}_i$  and  $\mathbf{o}_j$  as well as the rest of pedestrians  $\tilde{\mathbf{p}}_m^{t+\Delta t}$ , and compute the repulsion acceleration each pedestrian has on  $i$ . The only pedestrians that have this repulsion effect on subject  $i$  are the ones which do not belong to the same group as  $i$  and  $\|\tilde{\mathbf{p}}_i^{t+\Delta t} - \tilde{\mathbf{p}}_m^{t+\Delta t}\| \leq 1m$ . The different avoidance terms are combined linearly.

Now the prediction of the pedestrian’s next position is also influenced by the avoidance term (acceleration) from all pedestrians:

$$\tilde{\mathbf{p}}_i^{t+\Delta t} = \mathbf{p}_i^t + (\mathbf{v}_i^t + \mathbf{a}_i^{t+\Delta t} \Delta t) \Delta t \quad (16)$$

The distance between prediction and real measurements is used to compute the cost:

$$C_{\text{SFM},i,j} = -\log E \left( \frac{\|\tilde{\mathbf{p}}_i^{t+\Delta t} - \mathbf{p}_j^{t+\Delta t}\|}{\Delta t}, V_{\text{max}} \right) \quad (17)$$

where the function  $E$  is detailed in Eq. [10](#).

In Figure 6 we plot the probability distribution computed using different terms. Note, this is just for visualization purposes, since we do not compute the probability for each point on the scene, but only for the positions where the detector has fired. There are 4 pedestrians in the scene, the purple one and 3 green ones walking in a group. As shown in 6(b), if we only use the predicted positions (yellow heads) given the previous speeds, there is a collision between the purple pedestrian and the green marked with a 1 collide. The avoidance term shifts the probability mode to a more plausible position.

### 3.2 Group Model

The social behavior [21] also includes an attraction force which occurs when a pedestrian is attracted to a friend, shop, etc. We model the attraction between members of a group. Before modeling group behavior we determine which tracks form each group and at which frame the group begins and ends (to deal with splitting and formation of groups). The idea is that if two pedestrians are close to each other over a reasonable period of time, they are likely to belong to the same group. From the training sequence in [10], we learn the distance and speed probability distributions of the members of a group  $P_g$  vs. individual pedestrians  $P_i$ . If  $m$  and  $n$  are two trajectories which appear on the scene at  $t = [0, N]$ , we compute the flag  $G_{m,n}$  that indicates if  $m$  and  $n$  belong to the same group.

$$G_{m,n} = \begin{cases} 1, & \sum_{t=0}^N P_g(m, n) > \sum_{t=0}^N P_i(m, n) \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

For every observation  $\mathbf{o}_i$ , we define a group label  $g_i$  which indicates to which group the observation belongs to, if any. If several pedestrians form a group, they tend to keep a similar speed, therefore, if  $i$  belongs to a group, we can use the mean speed of all the other members of the group to predict the next position for  $i$ :

$$\tilde{\mathbf{p}}_i^{t+\Delta t} = \mathbf{p}_i^t + \sum_{g_m=g_i} \mathbf{v}_m^t \Delta t \quad (19)$$

The distance between this predicted position and the real measurements is used in (10) to obtain the cost for the grouping term.

An example is shown in Figure 6(c), where we can see that the maximum probability provided by the group term keeps the group configuration. In Figure 6(d) we show the combined probability of the distance and SFM information, which narrows the space of probable positions. Finally, Figure 6(e) represents the combined probability of DIST, SFM and GR. As we can see, the space of possible locations for the purple pedestrian is considerably reduced as we add the social and grouping behaviors, which means we have less ambiguities for data association. This is specially useful to decrease identity switches as we present in Section 5.

## 4 Implementation Details

To compute the SFM and grouping costs, we need to have information about the velocities of the pedestrians, which can only be obtained if we already have the trajectories. We solve this chicken-and-egg problem iteratively as shown in Algorithm 1; on the first iteration, the trajectories are estimated only with the information defined in Section 2.2 for the rest of iterations, the SFM and GR is also used. The algorithm stops when the trajectories do not change or when a maximum number of iterations  $M_i$  is reached.

---

### Algorithm 1. Iterative optimization

---

```

while  $\mathcal{T}_i \neq \mathcal{T}_{i-1}$  and  $i \leq M_i$  do
  if  $i == 1$  then
    1.1. Create the graph using only DIST information
  else
    1.2. Create the graph using DIST, SFM and GR information
  end if
  2. Solve the graph to find  $\mathcal{T}_i$ 
  3. Compute velocities and groups given  $\mathcal{T}_i$ 
end while

```

---

**Linear Programming Solvers.** The minimum cost solution is found using the Simplex algorithm [30], with the implementation given in [31]. Though Simplex has an exponential worst-case complexity, we are able to track most sequences in just a few seconds; this is because each node represents one detection, and therefore the dimension of the graph is quite small. For larger graphs [9] or more crowded environments, we can use the k-shortest paths solver [9, 32] which has a worst case complexity of  $O(k(m + n \cdot \log(n)))$ . For more details on network flows and Simplex we refer the reader to [33], and to [34] for more information on the k-shortest path algorithm.

**Integrity of the Solution.** When defining the program to be solved, we saw that Eq. (7) defined an integer program, which is known to be NP-complete. We relaxed the condition into Eq. (8) in order to use efficient Linear Programming solvers to find the optimum solution to our problem. If the solution to the relaxed version of the program is integer, then we know it is an optimal solution of the original problem [33]. The question is, can we guarantee that the solution will be always integer?

Let us assume the conditions of the Linear Program are expressed as:  $Ax = b$ . If all entries of  $A$  and  $b$  are integer, as it is our case, we can determine that  $Ax = b$  has an integer solution by Cramer's rule:

$$Ax = b \quad \Leftrightarrow \quad x = A^{-1}b \quad \Leftrightarrow \quad \forall i : x_i = \frac{\det(A^i)}{\det(A)} \quad (20)$$

where  $A^i$  is equal to  $A$  except on the  $i$ -th column where it is equal to  $b$ . From here, we can determine that  $x$  will be integer when  $\det(A)$  is equal to  $+1$  or  $-1$ . A matrix  $A \in \mathbb{Z}^{m \times n}$  is *totally unimodular* if the determinant of all the sub-square matrices of  $A$  is either  $0$ ,  $+1$  or  $-1$ .

*Theorem 1:* If  $A$  is totally unimodular, every vertex solution of  $Ax \leq b$  is integer.

A well-known case of totally unimodular matrices are the node arc incidence matrices  $N$  of a directed network. Therefore, our defined constraint matrix is totally unimodular, and the solutions we will obtain will always be integer.

**Computationally Reduction.** To reduce the computational cost, we prune the graph using the physical constraints represented by the edge costs. If any of the costs  $C_{ij}$ ,  $C_{\text{SFM},i,j}$  or  $C_{\text{GR},i,j}$  is infinite, the two detections  $i$  and  $j$  are either too far away to belong to the same trajectory or they do not match according to social and grouping rules, therefore the edge  $(i, j)$  is erased from the graphical model. For long sequences, we divide the video into several batches and optimize for each batch. For temporal consistency, the batches have an overlap of  $F_{\text{max}} = 10$  frames. With our non-optimized code, the runtime for a sequence of 800 frames (114 seconds), 4837 detections, batches of 100 frames and 6 iterations is 30 seconds on a 3GHz machine.

## 5 Experimental Results

In this section we show the tracking results of our method on three publicly available datasets and compare with existing state-of-the-art tracking approaches using the CLEAR metrics [35], which split the measuring scores into *accuracy* and *precision*:

- **Detection Accuracy (DA):** measures how many detections were correctly found and therefore is based on the count of missed detections  $m_t$  and false alarms  $f_t$  for each frame  $t$ .

$$DA = 1 - \frac{\sum_{t=1}^{N_f} m_t + f_t}{\sum_{t=1}^{N_f} N_G^t}$$

where  $N_f$  is the number of frames of the sequence and  $N_G^t$  is the number of ground truth detections in frame  $t$ . A detection is considered to be correct when it is found within 50 pixels from the ground truth and the bounding boxes of both ground truth and detection have some overlap.

- **Tracking Accuracy (TA):** similar to DA but also including the identity switches  $i_t$ . In this case, the measure does not penalize identity switches as much as a missing detection or a false alarm as we use a  $\log_{10}$  weight.

$$TA = 1 - \frac{\sum_{t=1}^{N_f} m_t + f_t + \log_{10}(1 + i_t)}{\sum_{t=1}^{N_f} N_G^t}$$



- **Detection Precision (DP):** precision measurements represent how well the bounding box detections match the ground truth. For this, an overlap measure between bounding boxes is used:

$$Ov^t = \sum_{i=1}^{N_{\text{mapped}}^t} \frac{|G_i^t \cap D_i^t|}{|G_i^t \cup D_i^t|}$$

where  $N_{\text{mapped}}^t$  is the number of mapped objects in frame  $t$ , i.e., the number of detections that are matched to some ground truth object.  $G_i^t$  is the  $i$ th ground truth object of frame  $t$  and  $D_i^t$  the detected object matched to  $G_i^t$ . The DP measure is then expressed as:

$$DP = \frac{\sum_{t=1}^{N_f} \frac{Ov^t}{N_{\text{mapped}}^t}}{N_f}$$

- **Tracking Precision (TP):** measures the spatiotemporal overlap between ground truth trajectories and detected ones, taking into account also split and merged trajectories.

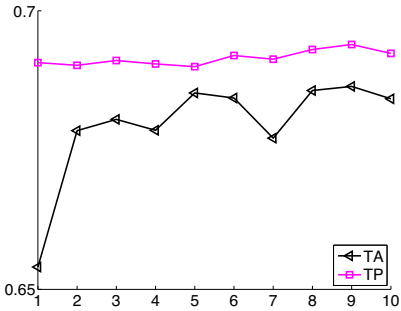
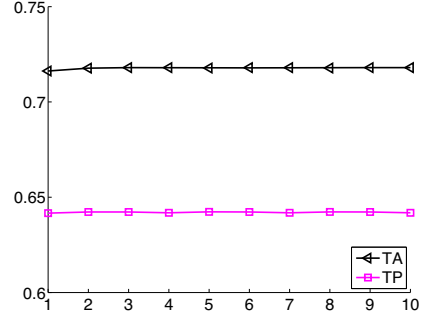
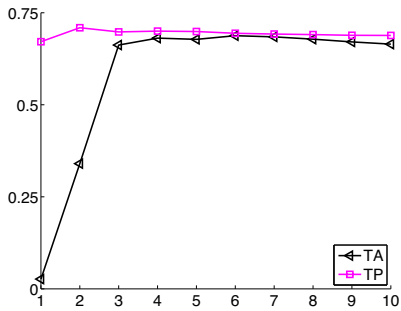
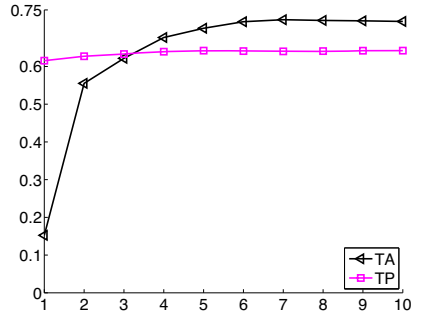
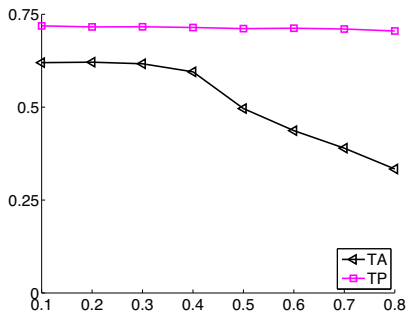
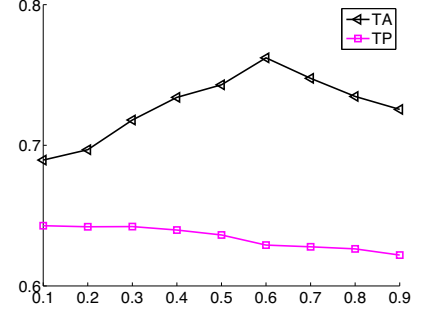
$$TP = \frac{\sum_{i=1}^{N_{\text{mapped}}^t} \sum_{t=1}^{N_f} \frac{|G_i^t \cap D_i^t|}{|G_i^t \cup D_i^t|}}{\sum_{t=1}^{N_f} N_{\text{mapped}}^t}$$

All experiments except the ones in Section 5.1 are performed with 6 iterations, a batch of 100 frames,  $V_{\text{max}} = 7m/s$ ,  $F_{\text{max}} = 10$ ,  $\alpha = 0.5$  and  $B_j = 0.3$ .

## 5.1 Analysis of the Effect of the Parameters

All parameters defined in previous sections are learned from training data; in our case we use one sequence of the publicly available dataset [10]. In this section we study the effect of the few parameters needed in our implementation, and show the proposed graph works well for a wide range of these parameters and therefore no parameter tuning is needed to obtain a good performance. The analysis is done on two publicly available datasets: a crowded town center [36] and the well-known PETS2009 dataset [37], to see the different effects of each parameters on each dataset.

**Number of Iterations.** The first parameter we analyze is the number of iterations  $M_i$  that we allow. This determines how many times the loop between computing social forces and computing trajectories is performed as explained in Algorithm 1. Looking at the results on the PETS 2009 dataset in Figure 7(b), we can see that after just 2 iterations the results remain very stable. Actually, the algorithm reports no changes in the

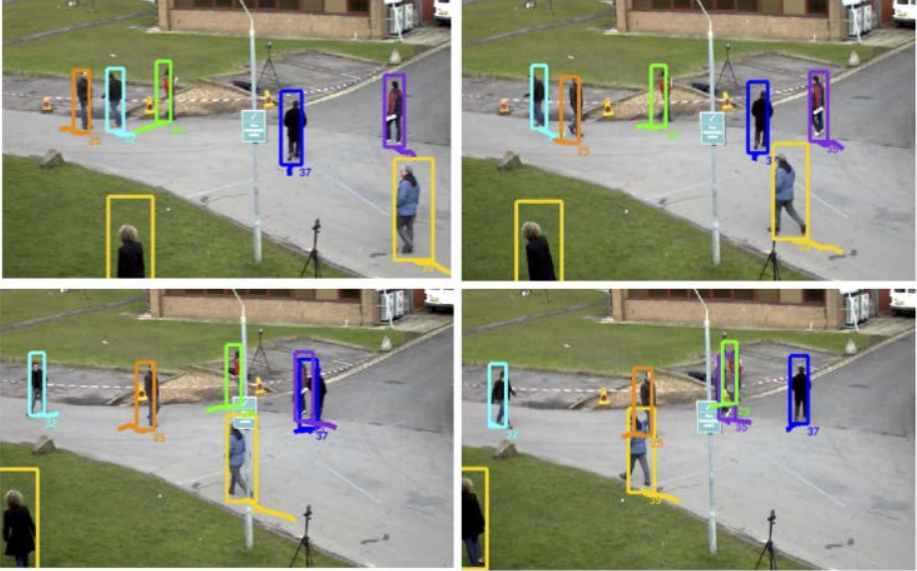
(a) TownCenter: iterations  $M_i$ (b) PETS2009: iterations  $M_i$ (c) TownCenter: maximum speed  $V_{max}$ (d) PETS2009: maximum speed  $V_{max}$ (e) TownCenter:  $B_j$ (f) PETS2009:  $B_j$ 

**Fig. 7.** Tracking accuracy (black) and precision (magenta) obtained for the Town Center dataset (left column) and the PETS 2009 dataset (right column) given varying parameter values

trajectories after 3 iterations, and therefore stops even though the maximum number of iterations allowed is higher. The result with 1 and 2 iterations is also not very different, which means the social and grouping behavior do not significantly improve the results for this particular dataset. This is due to the fact that this dataset is very challenging from a social behavior point of view, with subjects often changing direction and groups forming and splitting frequently. More details and comments on these results can be found in Section 5.3. On the other hand, we observe a different effect on the Town-Center dataset, shown in Figure 7(a). In this case, there is a clear improvement when using social and grouping behavior (i.e. the result improves when we use more than one iteration). We also observe a pattern on how the Tracking Accuracy of the dataset evolves: there is a cycle of 3 iterations for which the accuracy increases and decreases in a similar pattern. This means that the algorithm is jumping between two solutions and will not converge to neither one of them. This happens when pedestrians are close together for a long period of time but are not forming a group, which means that even with social forces, it is hard to say which paths they will follow.

**Maximum Speed.** This is the parameter that determines the maximum speed of the pedestrians that we are observing. In this case, we can see in Figures 7(c) and 7(d) a clear trend in which the results are very bad when we force the pedestrians to walk more slowly than they actually do, since we are artificially splitting trajectories. The results converge when the maximum speed allowed is around 3m/s - 5m/s, which is the reported mean speed of pedestrians in a normal situation. More interestingly, we observe that the results are kept constant when using higher maximum speed values. This is a positive effect of the global optimization framework, since we can use a much higher speed limit and this will still give us good results and will allow us to track a person running through the scene, a case of panic when people start running, etc.

**Cost for the Frame Difference.** The last parameter,  $B_j$ , appears in Eq. (12) and represents the penalization term that we apply when the frame difference between two detections that we want to match is larger than 1. This term is used in order to give preference to matches that are close in time. Here we can again see different effects on the two datasets. In Figure 7(e), we see that the results are stable until a value of 0.4. The lower the value, the higher is the penalization cost for the frame difference, which means it is more difficult to match those detections which are more than 1 frame apart. When the value of  $B_j$  is higher than 0.4, there are more ambiguities in the data association process because it is easier to match detections which are many frames apart. In the TownCenter dataset, there is no occluding object in the scene, which means missing detections are sporadic within a given trajectory. In this scenario, a lower value for  $B_j$  is better, since small gaps can be filled and there are less ambiguities. Nonetheless, we see different results in the PETS 2009 dataset in Figure 7(f), since here there is a clear occluding object in the middle of the scene (see Figure 8) which occludes the pedestrians for longer periods of time. In this case, a higher value of  $B_j$  allows to overcome these large gaps of missing data, and that is why the best value for this dataset is around 0.6.



**Fig. 8.** Four frames of the PETS2009 sequence (separation of 9 frames), showing several occlusions, both created by the obstacle on the scene and between pedestrians. All the occlusions can be recovered with the proposed method.

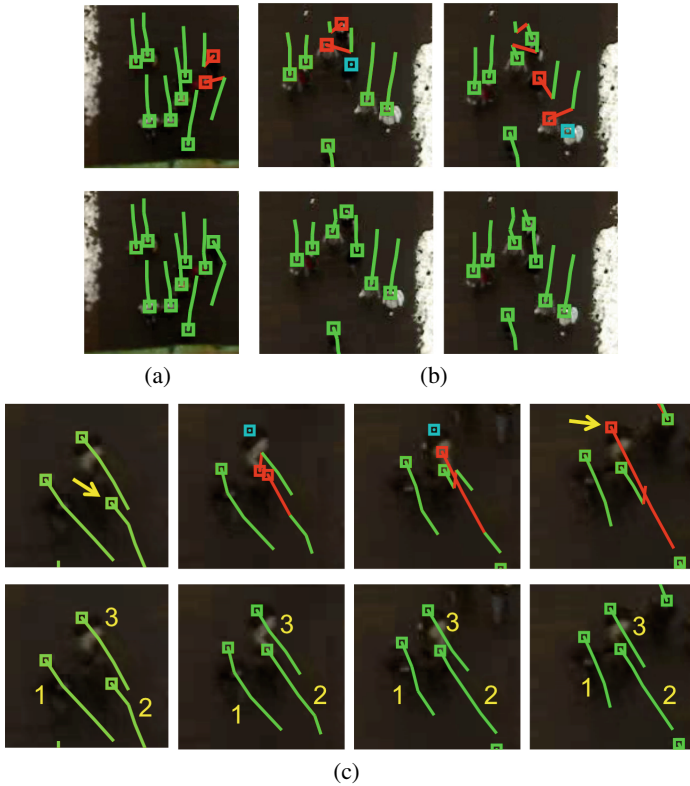
## 5.2 Evaluation with Missing Data, Noise and Outliers

We evaluate the impact of every component of the proposed approach with one of the sequences of the dataset [10], which contains images from a crowded public place, with several groups as well as walking and standing pedestrians. The sequence is 11601 frames long and contains more than 300 trajectories. First of all, we evaluate our group detection method on the whole sequence with ground truth detections: 61% are correctly detected, 26% are only partially detected, 13% are not found and an extra 7% groups are detected wrongly.

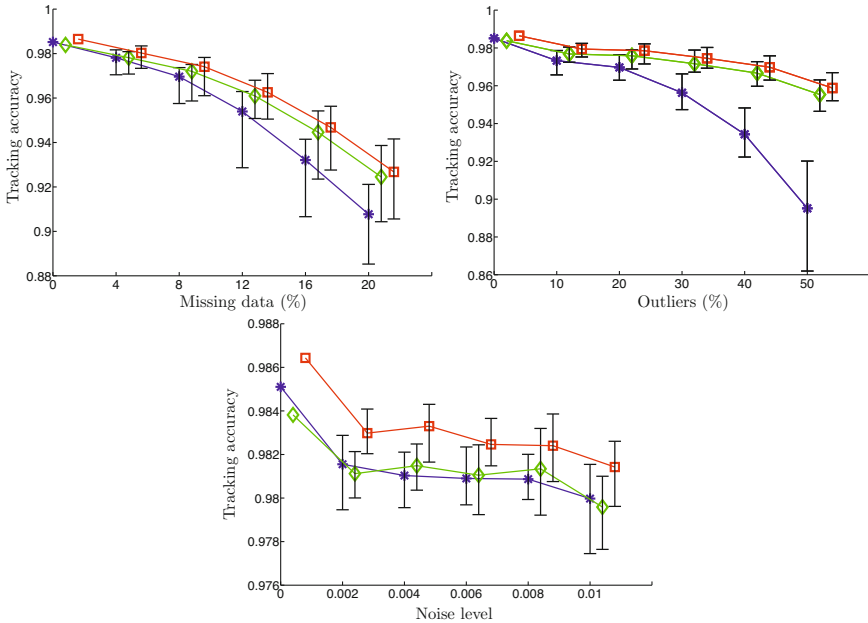
Using the ground truth (GT) pedestrian positions as the baseline for our experiments, we perform three types of tests, missing data, outliers and noise, and compare the results obtained with:

- DIST: proposed network model with distances
- SFM: adding the Social Force Model (Section 3.1)
- SFM+GR: adding SFM and grouping behavior (Section 3.2)

**Missing Data.** This experiment shows the robustness of our approach given missed detections. This is evaluated by randomly erasing a certain percentage of detections from the GT set. The percentages evaluated are [0, 4, 8, 12, 16, 20] from the total number of detections over the whole sequence. As we can see in Figure 10, both SFM and SFM+GR increase the tracking accuracy when compared to DIST.



**Fig. 9.** *Top row:* Tracking results with only DIST. *Bottom row:* Tracking results with SFM+GR. *Green* = correct trajectories, *Blue* = observation missing from the set, *Red* = wrong match. **9(a)** Wrong match with DIST, corrected with SFM. **9(b)** Missing detections cause the matches to shift due the global optimization; correct result with SFM. **9(c)** Missed detection for subject 3 on two consecutive frames. With SFM, subject 2 in the first frame (yellow arrow) is matched to subject 3 in the last frame (yellow arrow), creating an identity switch; correct result with grouping information.



**Fig. 10.** Experiments are repeated 50 times and average result, maximum and minimum are plotted. *Blue star* = results with DIST, *Green diamond* = results with SFM, *Red square* = results with SFM+GR. *From left to right:* Experiment with simulated missing data, with outliers, and with random noise.

**Outliers.** With an initial set of detections of GT with 2% missing data, tests are performed with [0, 10, 20, 30, 40, 50] percentage of outliers added in random positions over the ground plane.

In Figure 10, the results show that the SFM is especially important when the tracker is dealing with outliers. With 50% of outliers, the identity switches with SFM+GR are reduced 70% w.r.t the DIST results.

**Noise.** This test is used to determine the performance of our approach given noisy detections, which are very common mainly due to small errors in the 2D-3D mapping. From the GT set with 2% missing data, random noise is added to every detection. The variances of the noise tested are [0, 0.002, 0.004, 0.006, 0.008, 0.01] of the size of the scene observed. As expected, group information is the most robust to noise; if the position of pedestrian A is not correctly estimated, other pedestrians in the group will contribute to the estimation of the true trajectory of A.

These results corroborate that having good behavioral models becomes more important as the observations deteriorate. In Figure 9 we plot the tracking results of a sequence with 12% simulated missing data. Only using distance information can see identity switches as shown in Figure 9(a). In Figure 9(b) we can see how missing data affects

the matching results. The matches are shifted, this chain reaction is due to the global optimization. In both cases, the use of SFM allows the tracker to interpolate the necessary detections and find the correct trajectories. Finally, in Figure 9(c) we plot the wrong result which occurs because track 3 has two consecutive missing detections. Even with SFM, track 2 is switched for 3, since the switch does not create extreme changes in velocity. In this case, the grouping information is key to obtaining good tracking results. More results are shown in Figure 13, first row.

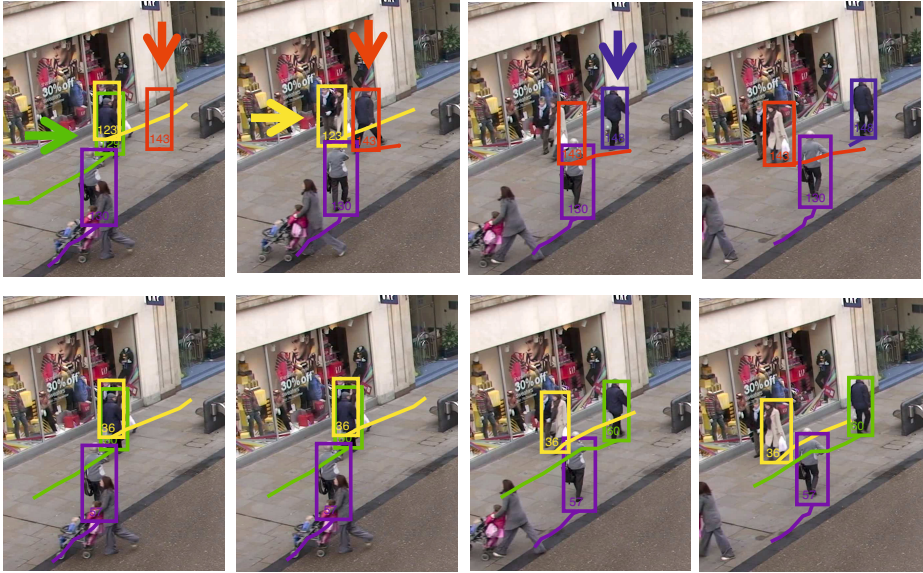


Fig. 11. Predictive approaches [10, 11] (first row) vs. Proposed method (second row)

### 5.3 Tracking Results

We evaluate the proposed algorithm on two publicly available datasets: a crowded town center [36] and the well-known PETS2009 dataset [37]. We compare results with:

- [36]: using the results provided by the authors for full pedestrian detections. The HOG detections are also given by the authors and used as input for all experiments.
- [8]: globally optimum tracking based on network flow linear programming, for which we use our own implementation.
- [10]: tracker based on Kalman Filter which includes social behavior, using the code provided by the authors.
- [11]: tracker based on Kalman Filter which includes social and grouping behavior, using our own implementation.

For a fair comparison, we do not use appearance information for any method. The methods [10, 11, 36] are online, while [8] processes the video in batches.

**Town Center Dataset.** We perform tracking experiments on a video of a crowded town center [36]. To show the importance of social behavior and the robustness of our algorithm at low frame rates, we track at 2.5fps (taking one every tenth frame). We show detection accuracy (DA), tracking accuracy (TA), detection precision (DP) and tracking precision (TP) measures as well as the number of identity switches (IDsw).

**Table 1.** Town Center sequence

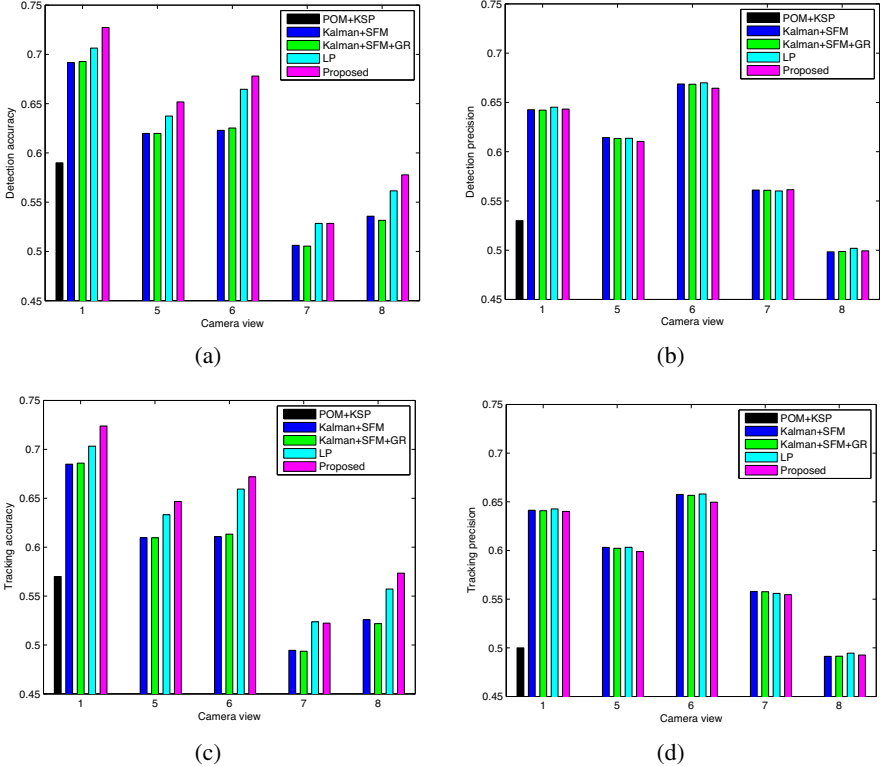
	DA	TA	DP	TP	IDsw
HOG Detections	63.1	–	71.9	–	–
Benfold et al. [36]	64.9	64.8	<b>80.5</b>	<b>80.4</b>	259
Zhang et al. [8]	66.1	65.7	71.5	71.5	114
Pellegrini et al. [10]	64.1	63.4	70.8	70.7	183
Yamaguchi et al. [11]	64.0	63.3	71.1	70.9	196
Proposed	<b>67.6</b>	<b>67.3</b>	71.6	71.5	<b>86</b>

Note, the precision reported in [36] is about 9% higher than the input detections precision; this is because the authors use the motion estimation obtained with a KLT feature tracker to improve the exact position of the detections, while we use the raw detections. Still, our algorithm reports 64% less ID switches. As shown in Table 1, our algorithm outperforms [10], which includes social behavior, and [11], which includes also grouping information, by almost 4% in accuracy and with 50% less ID switches. In Figure 11 we can see an example where [10, 11] fail. The errors are created in the greedy phase of predictive approaches, where people fight for detections. The red false detection in the first frame takes the detection in the second frame that should belong to the green trajectory (which ends in the first frame). In the third frame, the red trajectory overtakes the yellow trajectory and a new blue trajectory starts where the green should have been. None of the resulting trajectories violate the SFM and GR conditions. On the other hand, our global optimization framework takes full advantage of the SFM and GR information and correctly recovers all the trajectories. More results of the proposed algorithm can be seen in Figure 13, last row.

**Results on the PETS2009 Dataset.** In addition, we perform monocular tracking on the PETS2009 sequence L1, Views 1,5,6,7,8 and obtain the detections using the Mixture of Gaussians (MOG) background subtraction method. We compare the results with the previously described methods plus the monocular result of View 1 presented in [9], where the detections are obtained using the Probabilistic Occupancy Map (POM) and the tracking is done using k-shortest paths.

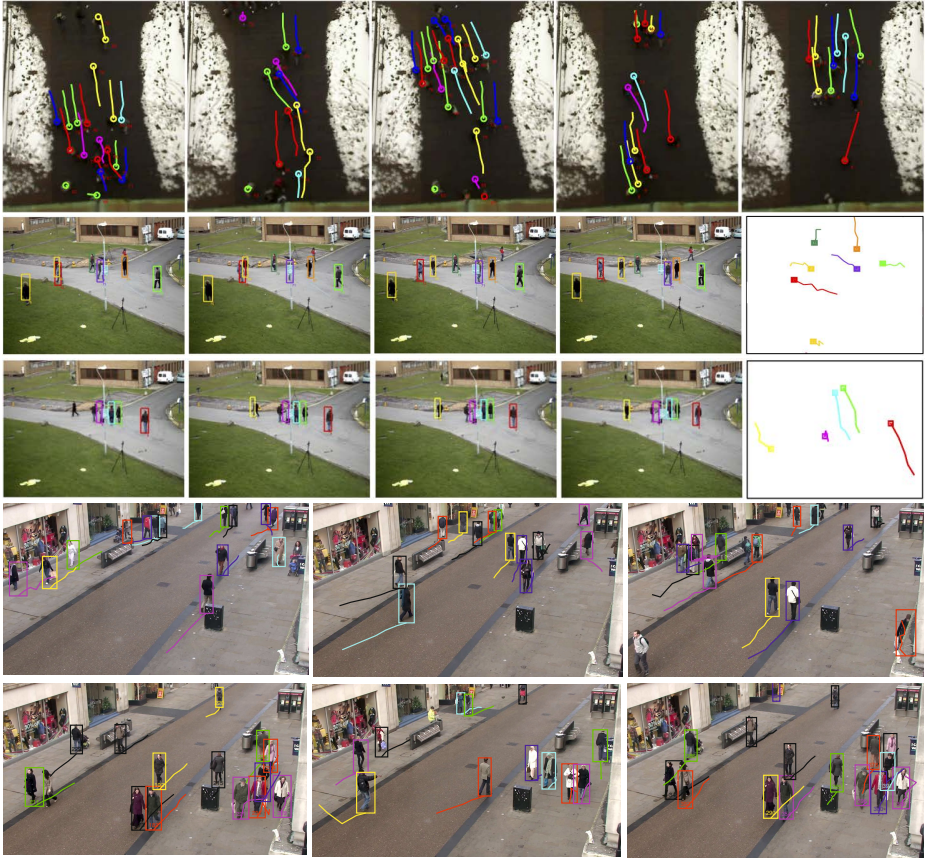
The first observation that we make is that the linear programming methods (LP and Proposed) clearly outperform predictive approaches in accuracy. This is because this dataset is very challenging from a social behavior point of view, because the subjects often change direction and groups form and split frequently. Since our approach is





**Fig. 12.** Results of the proposed method on the PETS2009 dataset views 1,5,6,7,8. (a) Detection accuracy, DA. (b) Detection precision, DP. (c) Tracking accuracy, TA. (d) Tracking precision.

based on a probabilistic framework, it is better suited for unexpected behavior changes (like destination changes), where other predictive approaches fail [10,11]. We can also see that the Proposed method has a higher accuracy in most views than the LP method, which does not take into account social and grouping behavior. The grouping term is especially useful to avoid identity switches between members of a group (see an example in Figure 13, third row, the cyan and green pedestrian who walk together). Precision is similar for all methods since the same detections have been used for all the experiments and we do not apply smoothing or correction of the bounding boxes. In general, views 7 and 8 are hard for tracking, due to 2D-3D calibration errors and a low field of view which means it is impossible to keep the identities and many small separate trajectories are created.



**Fig. 13.** *First row:* Results on the BIWI dataset (Section 5.2). The scene is heavily crowded, social and grouping behavior are key to obtaining good tracking results. *Second and third rows:* Results on the PETS2009 dataset (Section 5.3). *Last two rows:* Results on the Town Center dataset (Section 5.3).

## 6 Conclusions

In this paper, we argued for integrating pedestrian behavioral models in a linear programming framework. Our algorithm finds the MAP estimate of the trajectories total posterior including social and grouping models using a minimum-cost network flow with an improved novel graph structure that outperforms existing approaches. People interaction is persistent rather than transient, hence the proposed probabilistic formulation fully exploits the power of behavioral models as opposed to standard predictive and recursive approaches such as Kalman filtering. Experiments on three public datasets reveal the importance of using social interaction models for tracking in difficult conditions such as in crowded scenes with the presence of missed detections, false alarms and noise. We present an extensive analysis of the effect of the parameters to show the robustness of our method. Results show that our approach is superior to state-of-the-art

multiple people trackers. As future work, we plan on working on the optimization itself in order to find an efficient optimization method that keeps the linearity of the problem and at the same time does not require to iterate between computing the social forces and computing the data association. On the other hand, we also plan to extend our approach to even more crowded scenarios where individuals cannot be detected and therefore features might be used as in [38]. This will be a first step to bridge macroscopic and microscopic approaches for crowd analysis.

**Acknowledgements.** This work was partially funded by the German Research Foundation, DFG projects RO 2497/7-1 and RO 2524/2-1.

## References

1. Gall, J., Yao, A., Razavi, N., van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. TPAMI (2011)
2. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV (2009)
3. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. IJCV 75(2) (2007)
4. Leibe, B., Schindler, K., Cornelis, N., van Gool, L.: Coupled detection and tracking from static cameras and moving vehicles. TPAMI 30(10) (2008)
5. Kaucic, R., Perera, A., Brooksby, G., Kaufhold, J., Hoogs, A.: A unified framework for tracking through occlusions and across sensor gaps. In: CVPR (2005)
6. 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)
7. Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.: Data-driven crowd analysis in videos. In: ICCV (2011)
8. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR (2008)
9. Berclaz, J., Fleuret, F., Türetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. TPAMI (2011)
10. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: ICCV (2009)
11. Yamaguchi, K., Berg, A., Ortiz, L., Berg, T.: Who are you with and where are you going? In: CVPR (2011)
12. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: ICCV Workshops, 1st Workshop on Modeling, Simulation and Visual Analysis of Large Crowds (2011)
13. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. TPAMI (2005)
14. Leal-Taixé, L., Heydt, M., Rosenhahn, A., Rosenhahn, B.: Automatic tracking of swimming microorganisms in 4d digital in-line holography data. In: IEEE Workshop on Motion and Video Computing, WMVC (2009)
15. Nillius, P., Sullivan, J., Carlsson, S.: Multi-target tracking - linking identities using bayesian network inference. In: CVPR (2006)
16. Yang, M., Yu, T., Wu, Y.: Game-theoretic multiple target tracking. In: ICCV (2007)
17. Berclaz, J., Fleuret, F., Fua, P.: Robust people tracking with global trajectory optimization. In: CVPR (2006)

18. Jiang, H., Fels, S., Little, J.: A linear programming approach for multiple object tracking. In: CVPR (2007)
19. Andriyenko, A., Schindler, K.: Globally Optimal Multi-target Tracking on a Hexagonal Lattice. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 466–479. Springer, Heidelberg (2010)
20. Wu, Z., Kunz, T., Betke, M.: Efficient track linking methods for track graphs using network-flow and set-cover techniques. In: CVPR (2011)
21. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Physical Review E* 51, 4282 (1995)
22. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: CVPR (2009)
23. Pelechano, N., Allbeck, J., Badler, N.: Controlling individual agents in high-density crowd simulation. In: Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2007)
24. Scovanner, P., Tappen, M.: Learning pedestrian dynamics from the real world. In: ICCV (2009)
25. Luber, M., Stork, J., Tipaldi, G., Arras, K.: People tracking with human motion predictions from social forces. In: ICRA (2010)
26. Ge, W., Collins, R., Ruback, B.: Automatically detecting the small group structure of a crowd. In: WACV (2009)
27. Choi, W., Savarese, S.: Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 553–567. Springer, Heidelberg (2010)
28. Pellegrini, S., Ess, A., Van Gool, L.: Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 452–465. Springer, Heidelberg (2010)
29. Bertsekas, D.: *Nonlinear programming*. Athena Scientific (1999)
30. Dantzig, G.: *Linear programming and extensions*. Princeton University Press, Princeton (1963)
31. Makhorin, A.: Gnu linear programming kit (glpk) (2010), <http://www.gnu.org/software/glpk/>
32. Pirsivash, H., Ramanan, D., Fowlkes, C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR (2011)
33. Ahuja, R., Magnanti, T., Orlin, J.: *Network flows: Theory, algorithms and applications*. Prentice Hall (1993)
34. Suurballe, J.: Disjoint paths in a network. *Networks* 4, 125–145 (1974)
35. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation for face, text and vehicle detection and tracking in video: data, metrics, and protocol. *TPAMI* 31(2) (2009)
36. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: CVPR (2011)
37. Ferryman, J.: *Pets 2009 dataset: Performance and evaluation of tracking and surveillance* (2009)
38. Brostow, G., Cipolla, R.: Unsupervised detection of independent motion in crowds. In: CVPR (2006)

# An Evaluation Framework for Stereo-Based Driver Assistance

Nicolai Schneider<sup>1</sup>, Stefan Gehrig<sup>2</sup>, David Pfeiffer<sup>2</sup>, and Konstantinos Banitsas<sup>3</sup>

<sup>1</sup> IT-Designers GmbH, Esslingen, Germany

<sup>2</sup> Daimler AG, Team Image Understanding, Sindelfingen, Germany

<sup>3</sup> Brunel University, London, UK

`nicolai.schneider@it-designers.de,`  
`{stefan.gehrig,david.pfeiffer}@daimler.com,`  
`konstantinos.banitsas@brunel.ac.uk`

**Abstract.** The accuracy of stereo algorithms or optical flow methods is commonly assessed by comparing the results against the Middlebury database. However, equivalent data for automotive or robotics applications rarely exist as they are difficult to obtain. As our main contribution, we introduce an evaluation framework tailored for stereo-based driver assistance able to deliver excellent performance measures while circumventing manual label effort. Within this framework one can combine several ways of ground-truthing, different comparison metrics, and use large image databases.

Using our framework we show examples on several types of ground-truthing techniques: implicit ground truthing (e.g. sequence recorded without a crash occurred), robotic vehicles with high precision sensors, and to a small extent, manual labeling. To show the effectiveness of our evaluation framework we compare three different stereo algorithms on pixel and object level. In more detail we evaluate an intermediate representation called the *Stixel World*. Besides evaluating the accuracy of the Stixels, we investigate the completeness (equivalent to the detection rate) of the Stixel World vs. the number of phantom Stixels. Among many findings, using this framework enables us to reduce the number of phantom Stixels by a factor of three compared to the base parametrization. This base parametrization has already been optimized by test driving vehicles for distances exceeding 10000 km.

## 1 Introduction

Today's stereo and flow algorithms have reached a maturity level that allows their use in real-world systems. The development of efficient stereo algorithms is the first step in making vehicles able to recognize their surroundings and eventually drive themselves in the future. Unfortunately, the performance evaluation for such algorithms is still mostly limited to comparisons on the Middlebury database [32]. There, stereo and flow algorithms are benched against a few indoor images under controlled conditions. Most applications have to deal with a

---

<sup>1</sup> e.g. <http://vision.middlebury.edu/stereo/>

lot of different conditions which are not covered by such controlled data sets. Especially in the automotive field a limited sensitivity to adverse weather conditions is crucial. This requires a certain robustness of the applied algorithms. For such an outdoor imagery evaluation we need metrics to evaluate different algorithms or parameters and to compare their performance. The goal is to create a system that will automatically evaluate the computed 3D scene description of the environment. For this purpose, we introduce a performance evaluation framework considering the following three levels:

1. low-level: *pixel-level*, e.g. false stereo correspondences - based on stereo data where we use knowledge about object-free volumes to detect violations.
2. mid-level: *freespace/Stixel* [2], the object-free space in front of the car — the inverse is also called evidence grid/occupancy grid. This is computed directly from the stereo correspondences. The freespace forms a basis for many other object detection algorithms and thus is suitable for a mid-level evaluation. Similar, the Stixel World describes the objects limiting the freespace and is evaluated in detail here.
3. high-level: *leader vehicle measurement*. We pick one particular application where the leading vehicle is measured in front of the ego-vehicle. This data is needed for all adaptive cruise-control (ACC) variants. Depending on the implemented driver assistance function, different accuracy demands are needed for the distance, relative velocity, lateral position and width of the leading vehicle. We focus on the lateral position and width of the leader vehicle since we have a RADAR system that determines the distance and relative velocity very accurately and serves as ground truth for that part. The challenge for such applications is to create a correct object segmentation, and it is here that the choice of stereo algorithm becomes apparent.

Our evaluation framework working on these three levels covers the range of applications in which stereo is used in today’s automotive industry (e.g. [37]).

The structure of this paper is as follows: The related work on our system is detailed in Section 2. The basic framework for this analysis is described in Section 3. The ground truth needed to evaluate the tasks is introduced in the same section. To show the power of the evaluation framework we select several algorithms for evaluation that are described briefly in Section 4. In Section 5 more details on the used metrics to measure the performance are given. We have tested three different stereo algorithms on all evaluation levels of detail and show the results in Section 6.1. Evaluation results focusing on several aspects of the Stixel World are presented in Section 6.2.

## 2 Related Work

### 2.1 Evaluation of Computer Vision Algorithms

In the field of automotive, computer vision systems become increasingly powerful. Consequently, many driver assistance systems make use of them for. However,

under adverse weather conditions these systems do not possess the reliability required. Using image based sensor information for active braking or autonomous steering requires high safety levels, robustness, and accuracy with respect to the used algorithms. The more safety critical a system is the more effort has to be spent in the evaluation process of such vision algorithms. The correctness and the required integrity of these systems gain special importance when upcoming norms like ASIL (Automotive Safety Integrity Levels or ISO 26262) come into effect.

In [27], a general framework for performance evaluation of Computer Vision algorithms is presented, with a focus on object detection algorithms. However, all introduced metrics are limited to monocular sequences and to metrics within the image plane. Both methods are less relevant to robotics and driver assistance scenarios.

One of the major problems in evaluating computer vision algorithms is the generation of ground truth data against which results can be tested. Traditionally, most of the algorithms in literature are evaluated by measuring differences between the computed result and the Middlebury database [44]. However, for automotive applications this is not sufficient, because the automotive field is faced with a couple of challenges: Firstly, it has to deal with adverse weather and lighting conditions which are not covered by such controlled data sets. Secondly, the tremendously rising complexity of modern vision systems demand for new evaluation methods which cannot be performed on single images. In addition, a pixel-by-pixel comparison (as on Middlebury) is not applicable to sparse stereo or flow algorithms - an algorithm class that might serve driver assistance tasks very well.

In general, algorithms need to be tested on much larger datasets for obtaining statistically meaningful performance measure [9]. A step towards creating large ground truth datasets was made in [18]. The authors presented a reliable methodology for establishing a large database of ground truth data for a variety of sensors on mobile platforms. The goal was to publish large datasets to support other researchers to verify and evaluate their algorithms. An evaluation strategy for stereo algorithms on large amounts of images was also proposed in [36]. In that publication a performance evaluation scheme and corresponding metrics were suggested. The authors describe a method for producing low effort evaluation results without having real ground truth data. Some of the obtained results are reiterated in this research.

## 2.2 Ground-Truthing

In recognition tasks (e.g. [10,12]) manually annotated ground truth is widely used where Receiver-Operator-Curves (ROC), Precision-Recall-Curves, or classification rates are compared. There, ground-truthing is already necessary to provide the recognition algorithms with training data.

An example used to easily obtain some ground truth data is shown in [24], where an orthogonal method to determine the street plane is used to evaluate stereo algorithms. However, the street plane investigation only verifies small

parts of the image whereas for real automotive applications there are many other parts of the 3D scene which are of high importance.

In the current literature, several concepts have been used for generating ground truth data. Each of them have their corresponding advantages and drawbacks. The following sections will give a short overview of those concepts.

**Multi Sensor Technology.** Modern test vehicles are usually equipped with multiple sensors. LIDAR (Light Detection And Ranging), RADAR (Radio Detection and Ranging) and optical cameras are examples for those sensors. Using a multi sensor system has the advantage of detecting (or even compensating) for the various errors produced by each method yielding more reliable and accurate data. Different approaches have been found which propose an efficient fusion strategy as well as solutions in handling divergent data [11][7][42].

In [31] an evaluation strategy for the Stixel World was published using a high precision LIDAR as a reference sensor. The Stixel's distance information was compared against the LIDAR measurements. Different scenarios were recorded and the errors in various distances were analyzed. In order to realize the proposed concept with low effort, the technical challenges in synchronizing the different sensors were circumvented using the stop motion principle. Leaving, only simple scenarios (without any dynamic driving maneuvering) can be analyzed.

According to [31], Semi-Global Matching (SGM) and LIDAR behave differently to reflective vehicle objects like windows, mirrors or puddles. While the SGM stereo estimation smooths over these areas, the LIDAR looks right through those or even follows the reflected rays of light: an undesirable property of such a system. Consequently, using LIDAR as ground truth sensor makes an evaluation in these areas impossible.

Another evaluation example using several sensors of the same type was published in [28]. In this approach various common stereo matching algorithms were evaluated using three cameras. Two of them were used for the stereo matching and the third was used for reference in order to estimate the prediction error. By using metrics assessing the intensity differences of the first two cameras and comparing those with the output of the virtually computed third camera, it was possible to rate different stereo algorithms on real-world scenes.

**Manual Labeling Methods.** One of the most commonly used methods in generating ground truth data is the involvement of human expert interactions called *labeling*. As every application or algorithm has different requirements numerous approaches exist in designing ground-truthing tools. In general these can be categorized as automatic or semi-automatic ones [19]. The majority of the tools are semi-automatic as in most cases some additional information is needed for starting the ground truth extraction.

Tools supporting manual input often have the advantage that errors raised from model approximations or noisy data can be minimized through human verification and correction. These semi-automatic tools are not very efficient in generating large ground truth datasets as they involve human effort during the process.



Driver assistance imagery exhibit highly dynamic driving scenarios and often at least 50 frames are necessary in order to make a reliable statement on the performance of the applied algorithm. Consequently, labeling large amounts of sequences is time consuming. To overcome this problem some approaches were published incorporating available tracking mechanisms in ground-truthing tools [18]. Instead of labeling each frame from the beginning, trackers can be used to follow objects from one frame to the next so human inputs or corrections are only required if deviations occur [26].

**Synthetic Data.** Today a lot of effort is put in generating realistic synthetic scenes. Based on a physical model, static and moving objects are rendered and placed into a defined scenario.

Using synthetic sequences has the advantage that all parameters for every object are previously known. This accounts especially for the trajectories of the moving objects. Hence, an evaluation becomes simple because ground truth data can be calculated from ray-tracing principles and thus is available for every single image of the sequence. Moving the viewpoint of a virtual image makes it possible to generate image pairs for simulating stereo-vision and computing the ground truth disparity image.

The drawback of using synthetic scenes is the increased entropy of real life: it is next to impossible to create models for all the real-world situations. Adverse weather conditions such as rain, sun glare or snow are examples of that as their physical background is too complex for mapping it to a computer model.

Study [40] shows how synthetic scenes can have an negative impact on the performance of stereo and motion estimation. Their results show that optimizing algorithms for synthetic data can even make the results on real-world scenes worse. For example motion blur, weather, and exposure differences between the left and right image can highly influence the performance of the algorithms.

### 3 Evaluation Framework

The main aim of our evaluation framework is to provide an automatic method for evaluating and optimizing different stereo and flow algorithms over a large dataset [36]. By now, it has proved its strength to be well suited for all kind of image processing tasks. A large sequence database with more than 1500 sequences (200-400 frames per sequence) serves as input for the evaluation task. Since most of the vision algorithms consume a lot of computing power the idea is to write the raw data measurements into an Evaluation Database (EDB) and calculate the metrics afterwards. This has the advantage that a recalculation of our metrics can be done within seconds. Figure 1 shows an overview of the framework.

All algorithms that are tested perform their image processing tasks with a predefined parameter set on the stored test sequences. For a meaningful evaluation, the content of the database has to differ with respect to daytime (day, night), weather (rain, snow, fog), location and environment (city, rural roads,

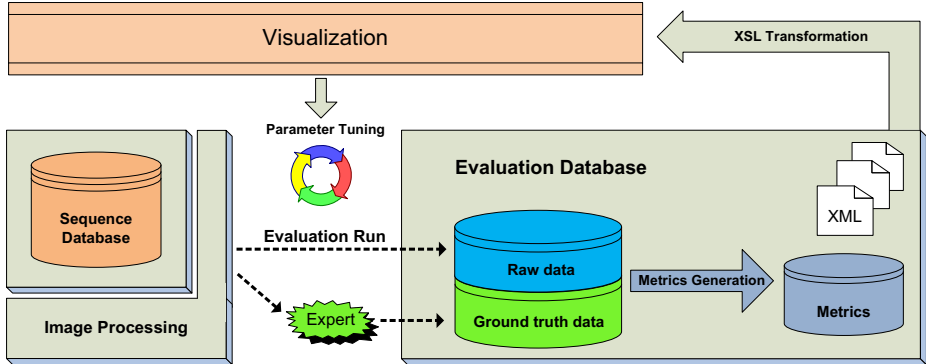
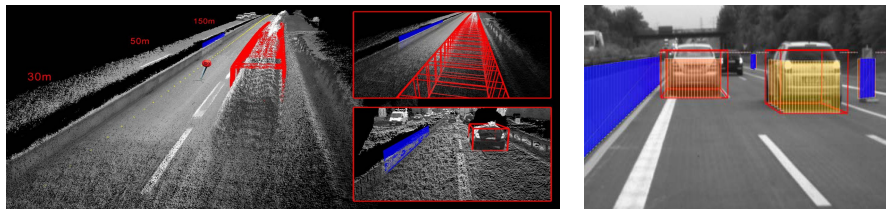


Fig. 1. Overview of the evaluation framework

highway). In an Evaluation Run (ER) for each frame of the sequence the measured raw data is written into the EDB. The ground truth data against which will be tested is either collected during a specific test run or defined manually by experts (manual ground-truthing). In case of manual ground-truthing, an appropriate software module is used providing manual interactions with the image. As a result of the image processing task a dataset with *ground truth* data and *measured* raw data is available in the EDB. A Metrics-Generator C++ module uses the generated datasets, computes the user defined metrics from it and saves it as an XML file back into the database. The processed data is visualized in a browser front-end by transforming the XML files with a predefined XSLT (<http://www.w3.org/TR/xslt>) style-sheet to SVG images. The transformation language (XSLT) provides an efficient strategy to transform a huge number of measurements into a few compact and easily explorable representations.

In addition, for each sequence a score is extracted by integrating the metrics frame-wise. By means of color encoded rankings one can easily determine those sequences which are relevant for further algorithm improvements. The user employs the sequence-wise accumulated metrics to choose candidates which could outperform the *current* ground truth. It takes only seconds in order to find and inspect relevant frames and to decide if the current candidate is a better ground truth or not. In order to verify the automatic testing process we use a subset of about 20 manually labeled ground truth datasets. A 3D editor and a tracking mechanism [4] allows effortless labeling of the scene infrastructure for this subset of sequences (see Figure 2). The accuracy of the manual ground truth is about 0.05 m error on average in the considered range (0 m - 40 m).

The 3D editor displayed in Figure 2 is used to create artificial ground truth data. For this purpose, static scene content from recorded sequences is projected into a virtual 3D view. Within that view, scene geometry is defined using basic geometrical shapes. During this step, dynamic scene content is taken into account by using the boxed-based tracking scheme proposed by Barth et al. [5,6].



**Fig. 2.** A 3D editor is used to manually create ground truth scene data. The right image shows the corresponding 2D output. The blue walls describe static scene infrastructure and the red boxes result from an object tracking algorithm to effectively evaluate moving objects.

An additional source of ground truth are robotic vehicles operated on a proving ground. For accuracy evaluation these robotic vehicles (having high-precision IMU) are used to perform predefined maneuvers repeatedly with high accuracy (errors  $< 0.02$  m). This results in accurately known position and motion states of the observed vehicles.

## 4 Algorithms Used

### 4.1 Stereo Algorithms

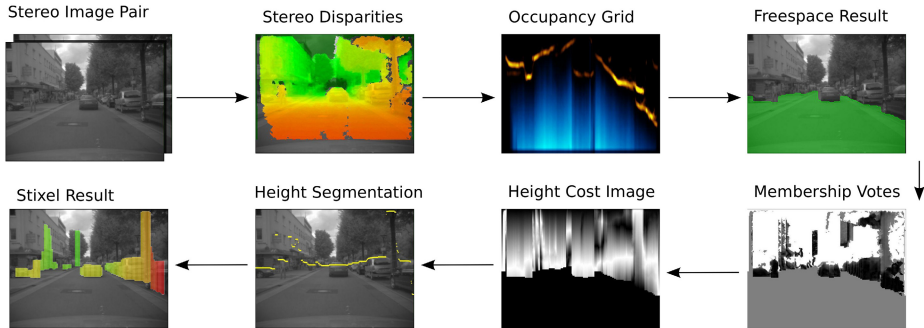
The initial motivation to build the evaluation system was in order to compare the following three stereo algorithms. All of these algorithms have real-time processing capability.

- *Signature-Based Stereo*: A signature based algorithm that searches for unique (corresponding) features of pixels [35].
- *Correlation Stereo*: A patch based correlation stereo algorithm using ZSSD (zero-mean sum of squared differences) [13].
- *Semi-Global Matching (SGM)*: Computes an approximated global optimum via multiple 1D paths in real-time [16].

### 4.2 Stixel World

The Stixel World [3,30] is a compact medium-level representation that describes the local three-dimensional environment. Stixels are defined as earthbound and vertically oriented rectangles with a fixed width (e.g. 5 px) and a variable height. Under these restrictions, Stixels are a 2.5D representation similar to Digital Elevation Maps [8]. From left to right, every obstacle within the image is approximated by a set of adjacent Stixels. This way, Stixels allow for an enormous reduction of the raw input data, e.g. 400.000 disparity measurements ( $1024 \times 440$  px stereo image pair) can be reduced down to only 200 Stixels.

Stixels simply give access to the most task-relevant information such as freespace and obstacles. For providing multiple independent vision-tasks with



**Fig. 3.** The Stixel World is extracted from stereo data in a cascade of multiple processing steps. This includes stereo matching, mapping stereo data to occupancy grids, freespace computation, a height segmentation and the final Stixel extraction step.

stereo-based measurement data, the Stixel World is neither too object-type specific nor too general and thus efficiently bridges the gap between low-level (pixel-based) and high-level (object-based) vision.

According to [3], Stixels are computed in a cascade of multiple processing steps: Mapping disparities to occupancy grids, a freespace computation, a height segmentation, and the final Stixel extraction step. For clarity, that process is visualized in Figure 3. Besides using Stixels to represent static environments, relying on the 6D-Vision [14] based Kalman filtering techniques allows for robustly tracking Stixels over time. Since the tracked objects are expected to move earth-bound, the estimated state  $\underline{X}$  is four-dimensional and consists of the lateral ( $X$ ) and longitudinal ( $Z$ ) position as well as the corresponding velocity components, such that  $\underline{X} = (X, Z, \dot{X}, \dot{Z})^T$ . As a result, motion information about the obstacles in the scene is available for every Stixel independently [30].

Making this work demands to have knowledge about the own motion state and requires to measure displacements of Stixels between two consecutive images. Ego-motion is provided by either visual odometry [1,15], SLAM [21,22,23] (self localization and mapping), or the inertial motion sensors of the vehicle. Stixel motion is obtained by computing optical flow correspondences. To achieve this a number of different methods are described in current literature. A short selection of those methods is listed below.

### 4.3 Optical Flow Schemes

Tracking Stixels over time in order to estimate the velocity of other moving obstacles requires the measuring of the two-dimensional displacement of these objects within the images of two consecutive time steps. This is achieved by computing the optical flow correspondences for exactly those areas.

Within the scope of this evaluation, four different flow methods have been chosen for testing. In the following, their particular differences, assets and drawbacks are highlighted briefly.

**Sparse KLT.** In [25], Lucas et al. suggest an optical flow scheme for feature tracking that relies on the gradient-based Lucas-Kanade method. The actual displacement is computed by solving the optical flow equations resulting from the constant brightness assumption for all the pixels in the neighborhood of a center point. This is achieved by means of a least squares error minimization.

Aiming at gaining robustness to global illumination changes, the matching criteria of this scheme is adapted to support a more robust measure, the zero-mean sum of squared differences (ZSSD).

A benefit of this method is the possibility to use the Kalman filter prediction of the tracked objects for initialization. This noticeably supports the estimation of large optical flow vectors and reduces effects resulting from texture ambiguities (e.g. repetitive patterns such as guard rails).

**Patch KLT.** The Patch KLT method is an extension of the KLT feature tracker to larger  $m \times n$  sized feature patches. In order to take perspective effects into account the change of scale is part of the estimation process. Additionally, an individual weight is considered for each pixel that is computed from the corresponding disparity measurement and the disparity of the tracked Stixel. This way, the influence of (background) pixels that lie within the patch (but do not belong to the actual tracked object) is minimized.

The Patch KLT benefits from leveraging texture information much better than competitive methods. Just like the sparse KLT method, the Patch KLT allows to be initialized with the prediction of the Kalman filter state.

**Census Optical Flow.** Stein [35] presents a method that allows to compute optical flow using the Census transform [41] as matching criteria. The census signatures are mapped to a hash table which is then used to determine optical flow correspondences between two images.

The benefit of this method is the constant run time independent of the maximum optical flow vector length. On the downside, this scheme does not allow to incorporate the motion state of the tracked object during initialization.

**Dense TV-L1 Optical Flow.** Müller et al. [29] have proposed a dense TV-L1-based method that puts dedicated focus on the application in open road scenarios. It incorporates additional stereo and odometry knowledge about the three-dimensional scenario. Their scheme is a variant of the work proposed by Zach et al. [43]. The implementation used does not consider information about the objects motion state for initialization.

## 5 Used Evaluation Metrics

Evaluating over large datasets demands effortless execution strategies and simple metrics which yield valuable information on the robustness and accuracy of an algorithm. Low-level metrics reflect the performance of a pixel-wise algorithm (e.g. the stereo matching scheme), mid-level metrics rate the quality of a possible intermediate representation at a later stage (e.g. the Stixel World) in the

data processing chain, and high-level metrics consider the object level. The used metrics are described in more detail in [33].

Typically, errors occur on sensor failures, atypical events (e.g. wipers crossing the windshield), or adverse weather and poor lighting conditions. Thus, for the purpose of our evaluation the following two aspects are examined:

- *Robustness*: represents the algorithm’s ability to deal with challenging situations like adverse weather and lighting conditions.
- *Accuracy*: describes the precision with which a Stixel represents the object in the real world.

## 5.1 Robustness

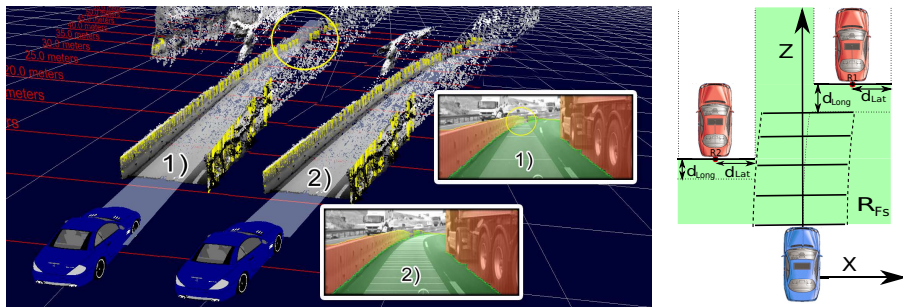
In the context of safety-critical vision-based driver assistance, the robustness of the used algorithms is of uttermost importance. With respect to robustness, it is reasonable to distinguish between algorithms operating on the pixel layer and those that use the object layer. For instance, a single error during stereo matching is rather unlikely to lead to a drastic automatic intervention of the driving car. However, the situation might be different for several false alarms in the medium-level representation.

Naturally, object occurrences in the driving corridor have a high priority, because those objects might lead to a critical change in driving. Hence, the evaluation primarily focuses on errors occurring within the driving corridor.

**False Stereo Correspondences (low-level).** When dealing with a large sequence database it is neither practical nor expedient to create ground truth data manually. This is especially true for disparity depth maps, as this method turns out to be a very time-consuming and hardly a feasible undertaking. In our research, a different strategy was chosen:

The vehicle’s driven path through the three-dimensional scene is reconstructed before the evaluation. This is achieved by looking ahead the vehicle’s odometry information (velocity and yaw rate) from the recorded sequence meta-data. It enables us to evaluate the false positive rates up to distances of 40 m. In case of having other moving vehicles in the scene, the actual freespace is additionally restricted by using an independent RADAR sensor (Continental ARS300 long range RADAR [34]). During this process, the RADAR is considered as ground truth and the RADAR results were checked visually by backprojecting the RADAR results into the image. For clarity, the described strategy is illustrated in Figure 4.

Given good visual conditions, no stereo measurements should fall into that volume. Hence, all stereo correspondences that do so are registered as potential errors of the stereo matching scheme. Following that strategy allows us to process many sequences without the need for human inspection or interaction. In return, that gives us the opportunity to evaluate very large sequence data bases with minimal effort.



**Fig. 4.** With a prediction of the ego vehicle’s current state it is not possible to detect the oncoming sharp right-hand bend early enough (marked with the yellow ellipse). That means that only a prediction around 25 m would be possible. Instead, by looking ahead the odometry information and RADAR information allows us to define the drivable freespace up to distances of 40 m. The diagram on the right side depicts such a freespace.

**False Positives (mid-level).** In order to test for false positive detections on the medium level, the same strategy (as above) is followed.

In terms of the Stixel representation, a false positive is defined as a Stixel detection that cannot be associated to an actual object in the real world. Thus, similar to detecting false stereo correspondences, all Stixel observations that lie within the driving path are considered as false detections.

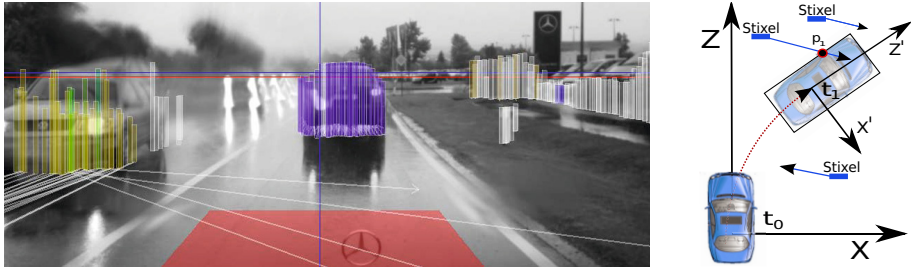
**Detection Range.** Another important characteristic for a vision system is the achieved detection range. Thus, for judging this property adequately, a so called completeness measure is defined. It reflects the detection rate of objects in the scene using ground truth object data.

For this evaluation task we use two different types of input data: Manually labeled sequences with a known 3D world geometry as well as robotic sequences. Robotic sequences correspond to driving scenarios with automated vehicles of which one carries the stereo system. Their motion path is known precisely by using iMAR iTrace-RT-F200 [20] IMUs as well as differential GPS.

For each time step in this database a corresponding ground truth Stixel representation is computed. A particular ground truth Stixel is considered as detected if a corresponding Stixel is computed from the input images (true positive). Consequently, both Stixels have to be within a depth-range of  $\pm 1$  m or  $\pm 3$  px disparities. Otherwise, the object is considered as missed (true negative). The corresponding completeness measure is defined as the ratio of the number of detected Stixels over the expected total amount of Stixels.

**Colliding Stixels.** In order to determine the robustness of the Stixel velocity estimate, it is preferable to have real-world ground truth data for all moving objects in a scene. Again, this is hard to achieve for a large dataset so instead of

performing a direct comparison, we use the Time To Collision (TTC) of a Stixel as an indicator for tracking errors. Since all of the scenarios in the database are recorded without having a collision, it can again be assumed that the TTC to other objects (static and moving) is greater than 1 s. Hence, if the predicted position of a Stixel intersects with the predicted vehicle position, a tracking error is registered. Figure 5 shows an exemplary inner city scenario. The red area visualizes the ego vehicle’s position within the next second. The arrows on the ground plane denote where the Stixels will move in that same period of time.



**Fig. 5.** Exemplary inner city scenarios with colliding Stixels. The red carpet indicates the ego vehicle’s position in the next second. The white arrows denote the Stixel positions in the same period of time. On the right the intersection check is visualized.

## 5.2 Evaluation of Accuracy

For accuracy evaluation the robotic vehicles are used. The vehicles perform pre-defined maneuvers. The IMU units record the exact paths of both platforms. The data is used for testing the accuracy of the distance measurement as well as the precision of the estimated velocity.

**Distance Error.** Both IMU units provide an accurate motion state for every frame. Using this data allows to transform all robotic motion states into the ego-system of the stereo camera rig used for testing. From that point onwards, it is straightforward to extract all Stixel measurements that are located on the other vehicle’s front and determine their mean distance so. This value is compared to the ground truth data of the IMU units.

**Velocity Error.** The evaluation of the velocity tracking error is split in two parts. Firstly, under the assumption that the current sequence is recorded in a static environment (i.e. without any moving objects), the mean absolute velocity error over all Stixel velocity estimates should equal to zero. Secondly, to evaluate while dealing with moving objects the robotic sequences are used. This way, the IMU velocity data is compared to the mean velocity estimate of those Stixels that represent the vehicles front in the image.



## 6 Evaluation Results

### 6.1 Stereo Performance

The sequences we used for stereo evaluation are divided into 50 % bad weather conditions (rain, snow, night) and 50 % normal conditions and contain a total of 22.100 frames recorded at 25 fps. The mixture is chosen to find failure modes of the algorithms as quickly as possible so less data will be needed.

The results in Table 1 show that the Signature-Based Stereo exhibits some shortcomings. Correlation Stereo is far better than Signature-Based Stereo, but the best method at all levels and metrics is SGM. The results from the bad weather part of the database are shown separately using parentheses.

Furthermore, the freespace computation and the leader vehicle measurement parameters were tuned for the Correlation Stereo method. For this reason, the obtained results underline the overall good and stable performance of SGM. If the applications were tuned with respect to SGM, the results of SGM would be even better. Especially the *availability* of the leader vehicle measurements outperforms the correlation approach by far. With the used freespace algorithm and metric at hand, we obtain similar results and the same ranking of stereo algorithms using the Stixels as intermediate representation.

**Table 1.** Evaluation result comparing census-stereo, correlation stereo, and SGM. SGM outperforms the other algorithms on all levels of detail.

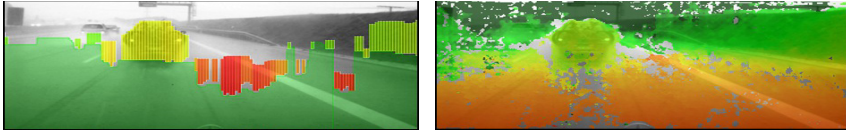
Metric	low-level	mid-level	high-level ( <i>LVM</i> )		
			<i>False Corr.</i>	<i>FS Diff.</i>	$\Delta Lat.Pos.$
Algorithm / Unit	$m_{fc}$ [%]	$m_{fs}$ [px]	$m_{lp}$ [cm]	$m_w$ [cm]	[%]
Signature-Based Stereo	7.45 (10.35)	3.04 (3.06)	19 (22)	26 (35)	80 (88)
Correlation Stereo	1.02 (1.47)	1.26 (1.72)	13 (15)	19 (37)	95 (99)
Semi-Global Matching	0.98 (0.94)	0.68 (0.79)	11 (12)	14 (32)	99 (99)

### 6.2 Stixel Robustness

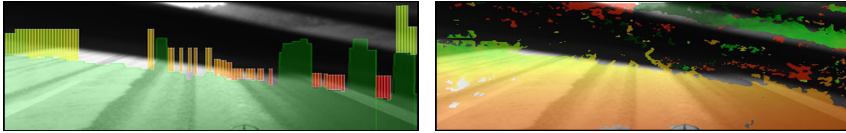
For the robustness evaluation of the Stixels, the complete database with more than 500 recorded sequences was used. It includes typical urban environments, rural roads and highway scenarios at different day times and weather conditions.

**Phantom Rate.** The Stixel phantom rate was determined in the categories *Sunshine*, *Night*, *Rain*, *Heavy rain* and *Snow* and is measured in phantoms per frame. Examples of challenging scenes with occurring phantoms are shown in Figure 6.

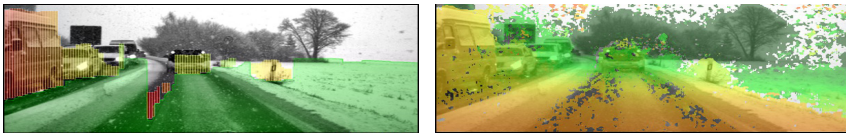
The results in Figure 7 primarily show that under optimal environmental conditions an excellent low error rate is achieved. However, this result change for adverse weather conditions such as *Rain* or even *Heavy Rain*, where the phantom rates are considerably higher. *Snow* on the other hand turns out to be



(a) Stixel phantoms in a rainy highway scenario.

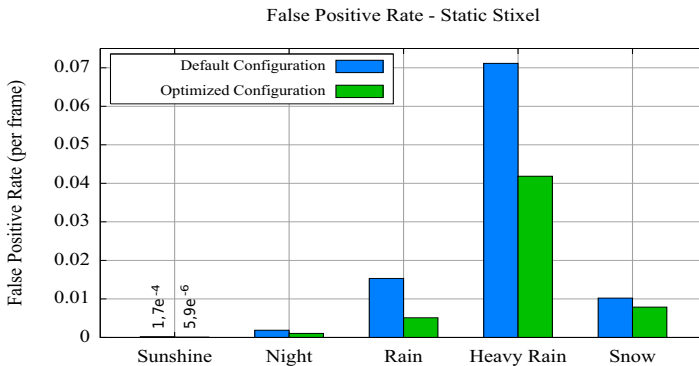


(b) Interference as a result of a wiper crossing the windscreen.



(c) Snow scenario with phantom Stixels.

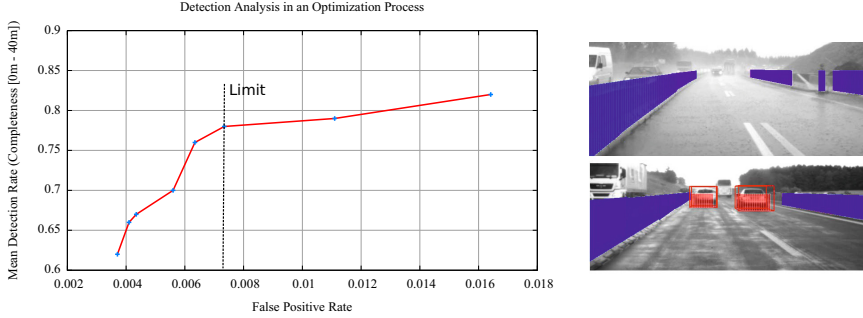
**Fig. 6.** The depicted Figures show different challenging scenarios of failure cases for the stereo computation and thus for the Stixel extraction. The visualization shows both the freespace/Stixel result as well as the disparity image.



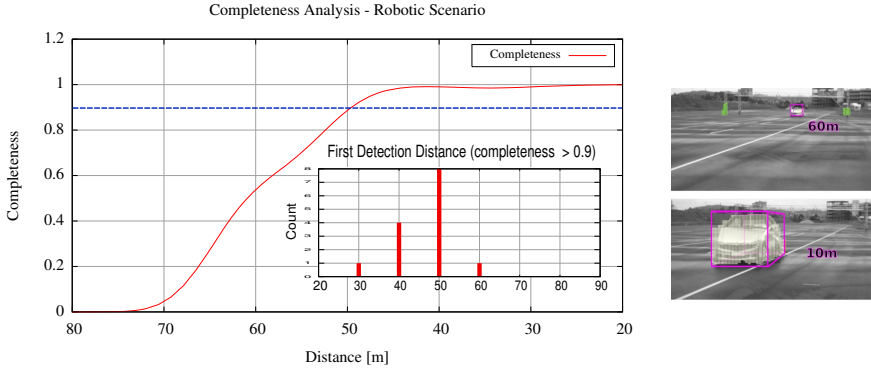
**Fig. 7.** The diagram depicts the Stixel phantom rates in the categories Sunshine, Rain, Heavy rain and Snow evaluated on a dataset exceeding 500 sequences

less of a problem than anticipated. This effect is mainly explained by the fact that, in contrast to rain, snow does not necessarily lead to a wet windshield and therefore does not cause a blurred sight.

In order to optimize for a low phantom rate, different parameters of the Stixel extraction schemes have been fine-tuned. At the same time it was important to consider the completeness metric. Otherwise, minimizing the phantom rate would inevitably lead to an arbitrary and possibly undesirable reduction of the



**Fig. 8.** The figure visualizes the completeness measure averaged over all labeled objects in the database. With an increasing optimization level, the number of phantoms decreases. However, a small phantom rate results in a low completeness. The diagram shows the limit after which further optimizations would downgrade the detection rate too much.



**Fig. 9.** The diagram shows the completeness in relation to the distance. The embedded histogram show the distribution of the distances at which the robotic vehicle reaches the 90 % completeness level for the first time.

object detection rate. In the sense of an ROC-curve, this dependency is visualized in Figure 8. The optimization was performed using manually labeled ground truth sequences with available 3D world geometry. This database consists of 20 manually labeled sequences with a total sum of approximately 1000 objects. The images in Figure 8 illustrate an extract of labeled database objects (red is moving, blue is static). In addition, the diagram depicts the limit up to which an optimization allows a robust object detection. A 100 % detection rate can not be reached due to violations of the assumed vertical pose constraint.

**Detection Range.** The detection range was evaluated on robotic scenarios. The priority was on that distance where the object detection exceeds 90 % completeness on the robotic vehicle. Consequently, for this purpose, only scenarios

with an oncoming vehicle covering a range of 0 m–80 m were of interest. Figure 9 shows the completeness over the distance. In order to have a meaningful result, the completeness is averaged over several sequences of the same type.

### 6.3 Tracking Performance - Testing Different Optical Flow Methods

Within Section 4.3 different Stixel tracking strategies have been discussed. This section aims to evaluate their performance and their quality with respect to the estimated motion states.

In terms of the object tracking, a core aspect is the computation of the Stixel displacement between the two consecutive images. For that task, we discussed multiple approaches that differ with respect to their technical prerequisites, their scope of action to combine the optical flow computation directly with the actual tracking process, and their computational effort. The Stixel tracking was tested in a stationary environment that contained no moving objects. Even though our own car was moving, the goal was to detect that the environment around us remains static.

The test using a static environment took place in a narrow urban environment with cars parked on both sides of the road. Naturally, the expectation for the motion state of all tracked objects is to have zero velocity. To stress the optical flow methods, the scenario was recorded several times while driving at different speeds, which includes 4, 8, 14 and 20 m/s ego-velocity. A snapshot of that sequence is depicted in Figure 10. The given figure also discusses different challenging aspects for the optical flow computation.

For estimating the optical flow between consecutive time steps the *Census-based* feature flow proposed by Stein [35], the dense *TV-L1* based optical flow scheme proposed by Müller [29], the *KLT-based* feature tracker proposed by Tomasi [39] and our own *Patch KLT* method were used. For the latter, a patch size of  $40 \times 16$  px (width $\times$ height) has proved a good working choice.

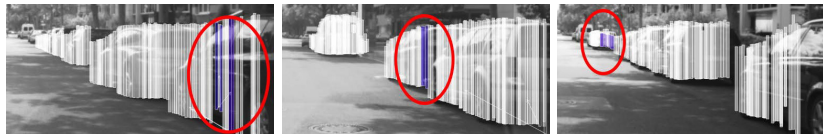
The results for the static environment are depicted in Figure 11. On the left side, for rating the tracking performance of the individual tracking schemes, the mean absolute velocity of all tracked Stixels is computed. Depending on the ego-velocity of the test vehicle, each sequence contributes about 300 to 1,000 frames. The evaluation is limited to a distance of 40 m.

Apparently, for the current setup, the different optical flow schemes are closely matched, such that there is no clear winner. Depending on the driven speed it is shown, that the mean velocity errors of all schemes rise with a linear characteristic. Yet, in reference to the total system complexity, that error is relatively small and lies between 6 % and 8 % of the driven ego-velocity. The obtained error curves seem plausible and match our expectations.

Altogether, the good performance of the investigated techniques is reasoned in the fact that the considered scenario is relatively simple. Thus, by changing to a highway-like environment, a more challenging scenario is taken into account. It features neither cars nor moving objects but has guard rails on both sides



(a) Correct estimation of the environment consisting of static objects. Thus, all Stixels are drawn with a white coloring which denotes a velocity close to zero.



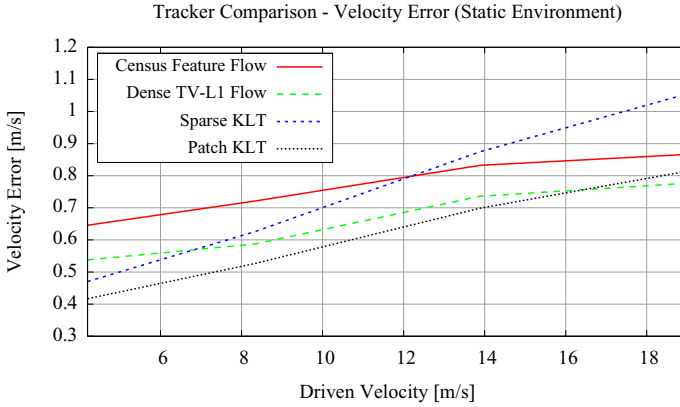
(b) Three typical sources for velocity errors during tracking within static environments are illustrated. The first figure shows a reflecting surface, the middle figure shows a jump in depth, and the third figure shows difficulties with motion estimation at large distances.

**Fig. 10.** Color visualizes motion. Ideally, all static objects should have a white coloring denoting zero velocity. This real-time color coding was used as quality indicator for the different tracking schemes. Figure (a) shows a good example, Figure (b) shows typical sources of error.

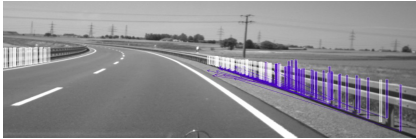
of the road. Naturally, due to their repetitive patterns, guard rails are likely to cause problems for the optical flow computation when driving along in parallel at high speeds. These problems are widely referred to as the *aperture problem* or the *blank wall problem* [7,38]. This is illustrated in Figure 12.

To increase the degree of difficulty, the ego-velocity is gradually increased to speeds of 8, 14, 20, 28 and 36 m/s. When looking at Figure 12a another important aspect becomes obvious. Problems within the optical flow estimation lead to holes within the line of Stixels covering the guard rail. Typically, this effect is caused by missing or erroneous optical flow measurements. Therefore, in order to draw a more practical conclusion, the performed tests included the completeness measure for the guard rail. This ratio is computed by using ground truth geometry. The corresponding evaluation results are shown in Figure 13.

Contrary to the previous more static test, the highway environment reveals severe differences between the tracking techniques. Depending on the particular tracking procedure, the velocity estimates as well as the detection rates vary noticeably. The best trade-off with respect to a low velocity error and a satisfying completeness measure is achieved by using the proposed *Patch KLT* procedure or dense *TV-L1* optical flow. Altogether, those two schemes are closely matched. In contrast, the *point feature based KLT* method and the *Census-based* optical flow tracking scheme have serious difficulties estimating the velocity correctly. The *sparse KLT* method yields a high completeness, but its mean absolute estimated velocity is unacceptably high when driving faster than 14 m/s. Even though the



**Fig. 11.** Direct comparison of the four different tracking schemes. Ideally, the mean absolute velocity should be zero. The quality of the optical flow measurement plays a significant role in this process. Thus, depending on the optical flow scheme, that goal is more or less achieved. For this static urban environment, the differences are rather small.



(a) Unreliable optical flow estimates on guard rails lead to wrong Stixel velocity estimates. Additionally, the guard rail is not covered completely.



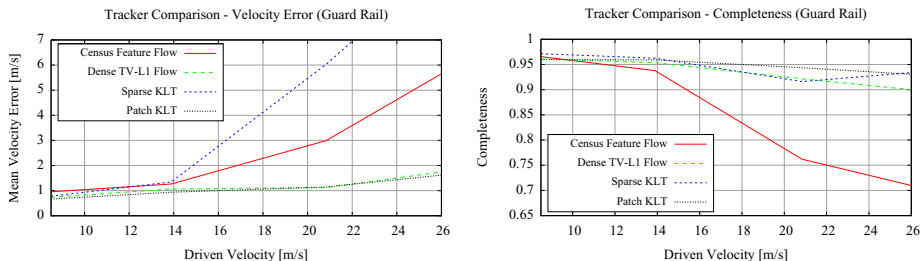
(b) In contrast, successful optical flow computation allows to obtain correct Stixel velocity estimates. The guard rail is covered much better.

**Fig. 12.** A precise optical flow estimate is essential for estimating the Stixel motion state reliably. Especially for structures that suffer from aperture problems at high ego velocities, this is a very challenging task. That matter is exemplified with a guard rail scenario.

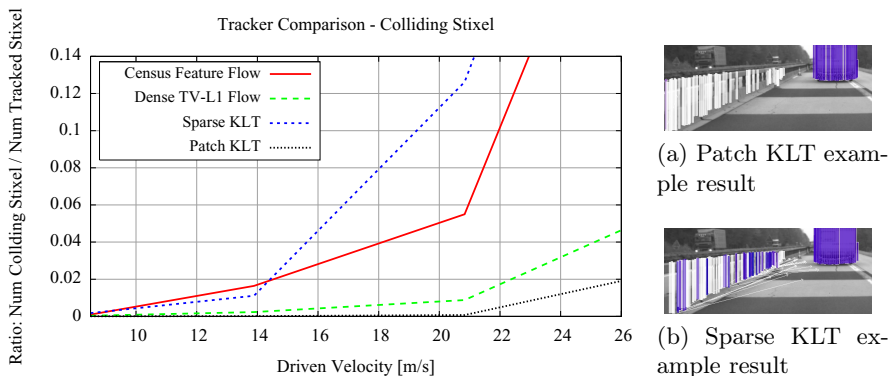
*Census-based* feature flow performs slightly better, the achieved velocity estimate is still not good enough to be used in terms of our objectives. Also, that flow scheme has severe problems regarding the detection rate. Thus, when going  $14 \text{ m/s}$  or faster, that ratio rapidly drops below 75 % completeness.

The good performance of the *TV-L1*-based optical flow is reasoned by the fact, that for every image the assumption of the world to remain static is used as a weak but apparently effective regularization prior for the optical flow estimation. Additionally, the globally optimizing property of *TV-L1* supports a solution that is smooth and thus supports our world model too.

With regard to the *Patch KLT*, things are quite similar. The used tracking scheme makes strong use of the Kalman filter prediction as a feed-forward signal. This clearly helps to resolve textural ambiguities of the tracked structures.



**Fig. 13.** This figure shows the results of the performance evaluation for the different tracking strategies using the guard rail scenarios. The left figure denotes the remaining mean absolute velocity for the different driven vehicle speeds (8, 14, 20, 28 and 36 m/s). Correspondingly, the right side shows the achieved completeness measure of Stixels covering the guard rail.



**Fig. 14.** The diagram shows a comparison of the four tracking approaches (Sparse-KLT, Patch KLT, Census Feature Flow and Dense TV-L1 Flow) in terms of colliding Stixels. The Patch KLT exhibits the fewest tracking errors.

This way, even though the *sparse feature based KLT* technique allows for the same procedure, things behave somewhat differently. For our understanding, the weakness of the *sparse KLT* method performance results from not considering the change of scale for the feature patch.

The proposed evaluation scheme is practicable as long as there are no moving objects within the scene. With respect to a robustness evaluation on larger datasets it is required to apply other metrics. Therefore, we use the number of colliding Stixels as indicator for tracking errors. Figure 14 demonstrates that the percentage of colliding Stixels correlates perfectly with the mean velocity error presented in the previous section.

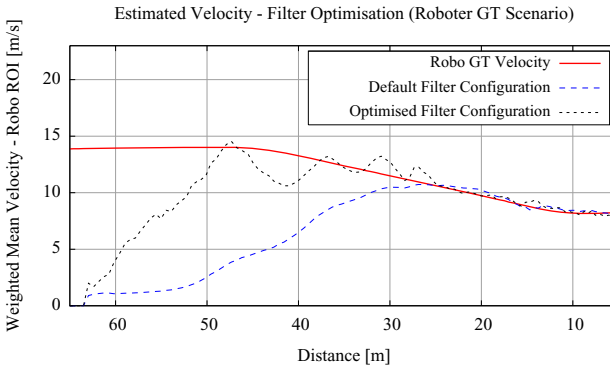
Finally, this allows us to evaluate the robustness of different tracking schemes under various weather conditions.

## 6.4 Stixel Accuracy

We use the robotic vehicle scenes to assess the Stixel velocity accuracy. The different flow algorithms performed similarly, hence we use the real-time Sparse KLT in the following scenarios. The accuracy of the Stixel measurements was evaluated on 30 robotic scenarios. Therefore, the defined metrics were analyzed within the scope of three different scenario types: Oncoming vehicles, turning maneuver and vehicles passing by.

**Velocity Error.** Robotic vehicles are used to obtain precise ground truth motion data. That data is used for testing the Stixel Kalman filter systems. For this evaluation all dynamic Stixels in the robotic vehicle ROI with an age greater than three frames were used. The resulting weighted mean velocity was compared to the robotic ground truth velocity. Hereby, the goal was to minimize the velocity error for the robotic sequences as well as for the static scenarios. Therefore, more than 20 different filter configurations have been tested.

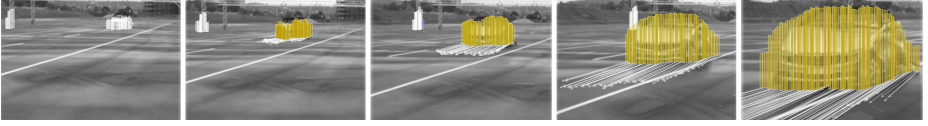
Figure 15 shows the resulting velocity estimation of an approaching vehicle before and after the optimization process. Both filter configurations perform similarly on the static scenes described in the previous section. The curves illustrate, that in contrast with the optimized filter configuration, the default filter configuration reaches the final velocity approximately 20 m later while exhibiting the same noise level on static scenes. The corresponding qualitative test results are shown in Figure 16.



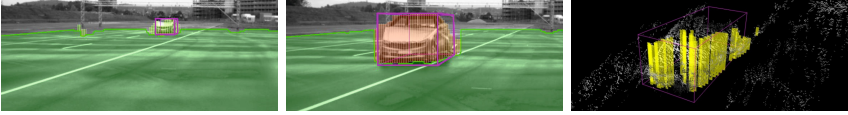
**Fig. 15.** This figure illustrates the velocity error for an oncoming robotic vehicle (c.f. Figure 16). The diagram shows the Kalman filtered velocity component of two different filter configurations. The ground truth velocity of the robotic vehicle is visualized in red. In contrast to the optimized filter configuration, the default filter configuration reaches the final velocity approximately 20 m later.

**Distance Error.** The distance error was evaluated for static and dynamic Stixel measurements on a variety of sequences with approaching robotic vehicles (see Figure 17). With the optimized filter configuration, both the measured and filtered Stixel’s distance information averaged over the vehicle front yield congruent output.



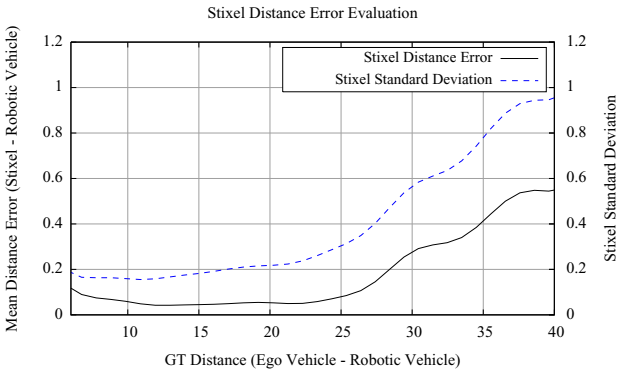


**Fig. 16.** Sequence of an approaching robotic vehicle. That vehicle starts at a distance of approximately 70 m and closely passes our vehicle to the left.



**Fig. 17.** Example images for the distance error evaluation. The calculated robotic vehicle position (marked in magenta) is projected into the image plane and used for collecting all the Stixel measurements representing the vehicle's front. On the right a 3D representation of the scene is visualized.

Figure 18 depicts the mean distance error to our robotic ground-truth distance of all static Stixels representing the front of the robotic vehicle. The second axis shows the calculated standard deviation for these Stixel measurements added on the mean error.



**Fig. 18.** The diagram shows the mean distance error of all Stixel measurements that are located at the front of the vehicle. The error increases in greater distances due to measurement noise as well as in the close-up range. The latter is explained by the violated vertical pose constraint on the engine hood. The 3D representation in Figure 17 visualizes the displaced Stixel position.

The curves show a noisy depth estimation at larger distances as well as an increasing distance error in the close-up range ( $< 15\text{ m}$ ). The measurements near the ego vehicle address the violated vertical pose constraint described in Section 4. As a consequence, if the vehicle's front with its engine hood is modeled end-to-end by one Stixel, its position in  $z$  direction will be displaced towards the

windshield (see Figure 17). That means, the Stixel measurements are seen as too far away. More details on the static Stixel position accuracy can be found in 31.

## 7 Conclusion

In this research we presented an evaluation framework for stereo-based driver assistance that operates on large image data bases and demands very little ground-truthing effort. To show the power of the evaluation framework, we performed evaluations on several stereo algorithms where we found the Semi-Global Matching (SGM) to be the best performing stereo algorithm on pixel-level, on freespace level and on object level. For the intermediate representation, the Stixel World, we detected Stixel phantoms only for challenging weather scenarios. By using the evaluation framework, the phantom rate could be further reduced by a factor of three while maintaining the detection rate of the Stixel representation. Comparing four optical flow algorithms used to generate dynamic Stixels we found the Patch KLT to be the best performing algorithm under the aspects of accuracy and robustness. For the absolute Stixel accuracy we determined a 0.5 m position error at 40 m distance using data from robotic vehicles as reference.

For future work we will extend this analysis framework to all vision-based driver assistance algorithms currently under development, to obtain meaningful performance figures. In addition, we consider making parts of the used data publicly available as part of a challenge that specifically addresses 3D outdoor scene analysis under all weather conditions.

## References

1. Badino, H.: A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform. In: Jähne, B., Mester, R., Barth, E., Scharf, H. (eds.) IWCM 2004. LNCS, vol. 3417, pp. 198–208. Springer, Heidelberg (2007)
2. Badino, H., Franke, U., Mester, R.: Free space computation using stochastic occupancy grids and dynamic programming. In: Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil (October 2007)
3. Badino, H., Franke, U., Pfeiffer, D.: The Stixel World - A Compact Medium Level Representation of the 3D-World. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2009. LNCS, vol. 5748, pp. 51–60. Springer, Heidelberg (2009)
4. Barth, A.: Vehicle Tracking and Motion Estimation Based on Stereo Vision Sequences. PhD thesis, Friedrich-Wilhelms-Universität zu Bonn (September 2010)
5. Barth, A., Franke, U.: Where will the oncoming vehicle be the next second? In: IEEE Intelligent Vehicles Symposium (IV), Eindhoven, Netherlands, pp. 1068–1073 (April 2008)
6. Barth, A., Siegemund, J., Franke, U., Förstner, W.: Simultaneous Estimation of Pose and Motion at Highly Dynamic Turn Maneuvers. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2009. LNCS, vol. 5748, pp. 262–271. Springer, Heidelberg (2009)

7. Brox, T., Weickert, J.: Nonlinear Matrix Diffusion for Optic Flow Estimation. In: Van Gool, L. (ed.) DAGM 2002. LNCS, vol. 2449, pp. 446–453. Springer, Heidelberg (2002)
8. Collins, R., Tsin, Y., Miller, J.R., Lipton, A.: Using a dem to determine geospatial object trajectories. In: Proceedings of the 1998 DARPA Image Understanding Workshop, pp. 115–122 (1998)
9. Courtney, P., Thacker, N., Clark, A.: Algorithmic modeling for performance evaluation. In: IAPR Conference on Machine Vision Applications (MVA), pp. 219–228 (1997)
10. Dreuw, P., Steingrube, P., Deselaers, T., Ney, H.: Smoothed Disparity Maps for Continuous American Sign Language Recognition. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 24–31. Springer, Heidelberg (2009)
11. Duffy, B.R., Garcia, C., Rooney, C.F.B., O’Hare, G.M.P.: Sensor fusion for social robotics. In: 31st International Symposium on Robotics, pp. 155–170 (2000)
12. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The 2005 pascal visual object classes challenge. Selected Proceedings of the 1st PASCAL Challenges Workshop, Springer (2006)
13. Franke, U.: Real-time stereo vision for urban traffic scene understanding. In: IEEE Intelligent Vehicles Symposium, IV (2000)
14. Franke, U., Rabe, C., Badino, H., Gehrig, S.: 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 216–223. Springer, Heidelberg (2005)
15. Fraundorfer, F., Scaramuzza, D., Pollefeys, M.: A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. In: IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA, pp. 1899–1904 (May 2010)
16. Gehrig, S.K., Eberli, F., Meyer, T.: A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 134–143. Springer, Heidelberg (2009)
17. Hohm, A., Wojek, C., Bernt, S., Winner, H.: Multi level sensorfusion and computer-vision algorithms within a driver assistance system for avoiding overtaking accidents. In: FISITA World Automotive Congress, pp. 1–14 (2008)
18. Hong, T., Chang, T., Takeuchi, A., Cheok, G., Scott, H., Shneier, M.: Performance evaluation of sensors on mobile vehicles using a large data repository and ground truth. In: Proceedings PerMIS (2003)
19. Huang, W., Tan, C.-L., Zhao, J.: Generating Ground Truthed Dataset of Chart Images: Automatic or Semi-automatic? In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) GREC 2007. LNCS, vol. 5046, pp. 266–277. Springer, Heidelberg (2008)
20. iMAR Navigation. iTraceRT-F200 (August 2011), <http://www.imar-navigation.de/>
21. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, pp. 486–492 (June 2010)
22. Lemaire, T., Berger, C., Jung, I.-K., Lacroix, S.: Vision-based slam: Stereo and monocular approaches. International Journal of Computer Vision (IJCV) 74(3), 343–364 (2007)

23. Levin, A., Szeliski, R.: Visual odometry and map correlation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, pp. 611–618 (June 2004)
24. Liu, Z., Klette, R.: Approximated Ground Truth for Stereo and Motion Analysis on Real-World Sequences. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 874–885. Springer, Heidelberg (2009)
25. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI 1981, Vancouver, Canada, pp. 674–679 (1981)
26. Manohar, V., Soundararajan, P., Raju, H., Goldgof, D., Kasturi, R., Garofolo, J.S.: Performance Evaluation of Object Detection and Tracking in Video. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 151–161. Springer, Heidelberg (2006)
27. Mariano, V.Y., Min, J., Park, J.H., Kasturi, R., Mihalcik, D., Li, H., Doermann, D.S., Drayer, T.: Performance evaluation of object detection algorithms. In: International Conference on Pattern Recognition, ICPR (2002)
28. Morales, S., Vaudrey, T., Klette, R.: Robustness evaluation of stereo algorithms on long stereo sequences. In: IEEE Intelligent Vehicles Symposium (IV), pp. 347–352 (2009)
29. Müller, T., Rannacher, J., Rabe, C., Franke, U.: Feature and depth-supported modified total variation optical flow for 3d motion field estimation in real scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, pp. 1193–1200 (June 2011)
30. Pfeiffer, D., Franke, U.: Efficient representation of traffic scenes by means of dynamic Stixels. In: IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, pp. 217–224 (June 2010)
31. Pfeiffer, D., Morales, S., Barth, A., Franke, U.: Ground truth evaluation of the Stixel representation using laser scanners. In: IEEE Conference on Intelligent Transportation Systems (ITSC), Madeira Island, Portugal (September 2010)
32. Scharstein, D., Szeliski, R.: Middlebury online stereo evaluation (2002), <http://vision.middlebury.edu/stereo>
33. Schneider, N.: Evaluation of stereo-based scene analysis under real-world conditions. Master's thesis, Brunel University (July 2011)
34. Continental Automotive Industrial Sensors. ARS 300 Long Range Radar Sensor 77 GHz (July 2011), [http://www.conti-online.com/generator/www/de/en/continental/industrial\\_sensors/themes/ars\\_300/ars\\_300\\_en.html](http://www.conti-online.com/generator/www/de/en/continental/industrial_sensors/themes/ars_300/ars_300_en.html)
35. Stein, F.: Efficient Computation of Optical Flow Using the Census Transform. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 79–86. Springer, Heidelberg (2004)
36. Steingrube, P., Gehrig, S., Franke, U.: Performance Evaluation of Stereo Algorithms for Automotive Applications. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 285–294. Springer, Heidelberg (2009)
37. Tech-News. Toyota' lexus ls 460 employs stereo camera, [http://techon.nikkeibp.co.jp/english/NEWS\\_EN/20060301/113832/](http://techon.nikkeibp.co.jp/english/NEWS_EN/20060301/113832/) (viewed April 15, 2009)
38. Tistarelli, M.: Multiple Constraints for Optical Flow. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 61–70. Springer, Heidelberg (1994)
39. Tomasi, C., Shi, J.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, pp. 593–600 (June 1994)

40. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: 23rd International Conference on Image and Vision Computing New Zealand, IVCNZ 2008, November 26-28, pp. 1-6 (2008)
41. Yamada, K., Mochizuki, K., Aizawa, K., Saito, T.: Motion Segmentation with Census Transform. In: Shum, H.-Y., Liao, M., Chang, S.-F. (eds.) PCM 2001. LNCS, vol. 2195, pp. 903-908. Springer, Heidelberg (2001)
42. Yilmaz, A.: Sensor fusion in computer vision. In: IEEE GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (2007)
43. Zach, C., Pock, T., Bischof, H.: A Duality Based Approach for Realtime TV-L<sup>1</sup> Optical Flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214-223. Springer, Heidelberg (2007)
44. Zanibbi, R., Blostein, D., Cordy, J.R.: White-box evaluation of computer vision algorithms through explicit decision-making (2009)

# Real-World Stereo-Analysis Evaluation

Sandino Morales, Simon Hermann, and Reinhard Klette

The *.enpeda..* Project, The University of Auckland, New Zealand  
pmor085@aucklanduni.ac.nz

**Abstract.** Evaluation of stereo-analysis algorithms is usually done by analysing the performance of stereo matchers on data sets with available ground truth. The trade-off between precise results, obtained with this sort of evaluation, and the limited amount (in both, quantity and diversity) of data sets, needs to be considered if the algorithms are required to analyse real-world environments. This chapter discusses a technique to objectively evaluate the performance of stereo-analysis algorithms using real-world image sequences. The lack of ground truth is tackled by incorporating an extra camera into a multi-view stereo camera system. The relatively simple hardware set-up of the proposed technique can easily be reproduced for specific applications.

## 1 Introduction

Vision-based *driver-assistance systems* (DAS) are designed to detect dangerous driving scenarios by understanding the 3-dimensional environment around the *ego-vehicle* (i.e. the mobile platform carrying the recording cameras). All the objects present in a given scene (e.g., other vehicles, pedestrians, road signs or the road itself) need to be detected and segmented, such that it can be decided whether they would represent a potential danger to the ego-vehicle. In this chapter, we are concerned about the evaluation of the depth estimated by using binocular stereo-matching algorithms.

Stereo-vision algorithms generate 3-dimensional information from a given scene by identifying corresponding pixels in (at least) a pair of images. Depth calculated via stereo-analysis algorithms is commonly incorporated into algorithmic pipelines as a basic step for a wide variety of applications (see, for example, [24,33]). Within the DAS context, stereo-analysis results contribute to different processes, such as object segmentation (e.g. pedestrians or other vehicles) [19], road modelling [35], or free-space detection [1].

Despite widespread acceptance of stereo-analysis algorithms as being a “fairly reliable” source of 3-dimensional data, there is still a need to develop an objective evaluation scheme that can evaluate their performance when using real-world images as input data. The lack of “true” measurements (i.e. for comparing with *ground truth*) represents a hard obstacle in this area, as exact camera pose

detection, together with the generation of precise 3-dimensional models of uncontrolled environments, is extremely difficult.<sup>1</sup>

In this chapter we discuss the evaluation method as presented in [30], we provide more detailed ways to compare the input and comparison data, discuss particular experiments, and summarise altogether our experience with this technique between 2009 and today. The basic idea of the technique is to use a third camera (the *control* camera) to evaluate the performance of binocular stereo-analysis: depth data calculated from the *reference* and *match* camera of a given stereo-camera system, are used to warp the image (say) of the reference camera into a *virtual image* that registers the scene as if it would be generated using the control camera. The virtual image is then compared with the actually recorded view in the control camera. (Of course, the 3-camera set-up can be generalised; the key-idea is to have one additional image for comparison).

In short, the technique offers an objective way to evaluate stereo-analysis algorithms using real-world data sets. The images of the control camera can be seen as being ground-truth data. The warping of the *reference image*, from a given stereo-pair into the virtual image, is defined by the calibrated camera geometry. We are not aiming at generating a “nice” warped image; we are just mapping intensity data of the reference image onto the nearest pixel in the image plane of the third camera (possibly overwriting previously mapped values). The control camera should also not be in a pose which supports similarities between virtual and *control image* (e.g. as it would be the case if the control camera would be positioned between reference and match camera).

The main advantage of the proposed evaluation technique is that the required hardware setup can be easily reproduced. Based on today’s time efficiency of stereo matchers, it can be used for real-time evaluations, and thus also as a basic module for designing an adaptive computer vision system for vision-based driver assistance (as discussed in [23]).

For our experiments we selected eight sequences of 400 *trinocular stereo sets* each (i.e. a stereo-pair plus the control image), recorded in different scenarios. The use of long sequences allows us to investigate the influence of changes in conditions when recording the stereo image data on the performance of the algorithms (e.g., local brightness variations between reference and match image, changes in scene geometry, camera issues, or lighting variations).

The remaining of this chapter is structured as follows. We start in Section 2 by reviewing some of the evaluation approaches found in the literature. In Section 3 we describe the generation of the virtual image and discuss the position of the control camera. This section also discusses the selected evaluation index. In Section 4 we briefly identify the stereo-analysis algorithms that are used for the presented experiments. For the selected trinocular sequences and a discussion about obtained evaluation results, see Section 5. Conclusions are provided in Section 6.

---

<sup>1</sup> The words *true* or *truth* are used in this chapter for a particular measuring method (e.g. manual measurements, or high-end laser-range data) considered to be “highly reliable”, but with being aware that measuring always involves errors.

## 2 Related Literature

Evaluation of stereo-analysis algorithms can currently be divided into two major groups. *Accuracy* is measured using data with available ground truth. *Confidence* is estimated for data recorded in real-world environments (without having ground truth available), for example by comparing stereo results of left-right and right-left matching.

Evaluation using data with available ground truth allows a precise comparison between the generated and the true values. But, it is limited by the quantity and diversity of available data sets. Test images, along with ground truth, are generated either in laboratories under highly controlled conditions (*engineered* images [34]), or by rendering 3-dimensional modelled scenes (*synthetic* images [39]).

Engineered images challenge algorithms with real-world objects that might be known as being problematic for stereo-analysis algorithms (e.g., textureless areas, slanted planes, and so forth). But, they are limited to a few images, showing close range scenarios that are almost free of real-world effects such as multiple light sources, non-Lambertian surfaces, unexpected shadows (lighting artefacts [40]), camera misalignment or blurring, and so forth. Scenes corresponding to common driving conditions (e.g. rainy days, busy pedestrian crossings, or different objects moving “randomly” and at multiple distances) cannot be recorded in a laboratory.

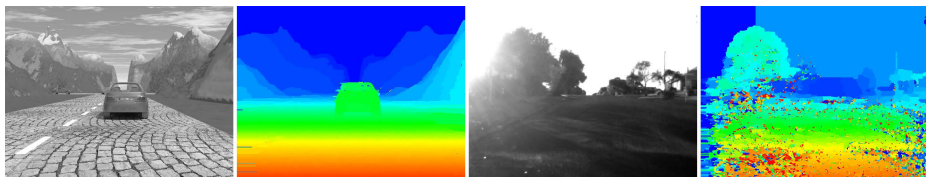
[34] presented an evaluation and classification scheme for stereo-analysis algorithms that has been widely followed by the computer-vision community. A main contribution of this work was a data set of several engineered stereo-pairs with available ground truth.

Synthetic data sets with available ground truth have also been made available online for some years, see for example [45]. The computer-generated data sets allow us to test the stereo-analysis methods in simulated environments where the algorithms are expected to work. However, synthetic data sets are limited by the models followed to generate the images (i.e. assuming pinhole-type cameras), and the motion of the objects (i.e. how is represented the motion of a walking pedestrian). Synthetic scenes are typically not yet aiming at a comprehensive physics-based modelling of cameras, lighting, or surfaces [23].

In the context of DAS, in [39] and [38] were presented data sets simulating “multi-second driving sequences” (e.g. of more than 50 stereo frames) with movement of both, the *virtual camera* and some of the objects present in the scene.

Ground truth-based evaluation is a good option for debugging, tuning of algorithms’ parameters, or for exploring new matching algorithms. For some applications, highly selective evaluations might be sufficient (i.e., for stereo-analysis of controlled environments such as automated factories or warehouses). But this cannot be expected for applications such as DAS, where stereo-vision programs have to provide reliable data on every road, under all kinds of weather conditions, and in any traffic context. According to [13], available data sets of engineered or synthetic images do only represent a very selective challenge for the algorithms,





**Fig. 1.** Disparity maps computed with the same stereo matcher (namely GCM-CEN, to be defined further below). *Left:* Reference image of a synthetic scene from [7], and computed disparity map. *Right:* Reference image of a real-world sequence and computed disparity map. Both disparity maps are encoded from red (maximum disparity) to blue (minimum disparity). GCM-CEN shows good performance on the synthetic sequence, but fails “totally” on the shown real-world sequence.

with different characteristics (formally defined in [13]) compared to real-world data.

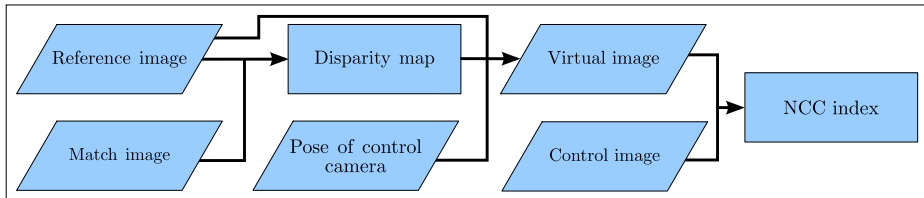
Figure 1 shows reference and disparity images for a synthetic (left) and a real-world (right) stereo-pair. Both disparity maps were generated with the same stereo algorithm (graph-cut stereo with census as cost function; see Section 4) using exactly the same parameters. For the “synthetic” disparity map it is easy to recognise all the objects present in the scene. For the real-world case, a lot of details are missing (e.g. the two trees on the left are merged into a single object) and a lot of noisy measurements are introduced.

One of the first evaluation schemes (data set and evaluation criteria) based on real-world stereo-pairs (without ground truth), was reported in [12]. The author provided twelve pairs of images to a selected number of research groups worldwide. For evaluating the calculated stereo measurements, manual checking was performed for around 50% of all the possible matchable pixels in the stereo-pairs.

A similar test bed was proposed in [3]. The authors made available to five research groups a ground truth-less data set of 49 stereo-pairs (the *JISCT* data set, visit [18]). Most of them are real-world images, but there are also engineered and synthetic stereo-pairs. However, none of them came with ground truth. The evaluation was based on a “reported value and unreported value” approach, i.e. whether the algorithm reported a value in (a manually) selected region where a measurement was feasible to be calculated.

Some other methods have been proposed to evaluate stereo-analysis algorithms in the absence of ground truth. In [8], the authors calculated (manually) true depth values at 200 randomly selected points. In [2], the evaluation was done by measuring the number of “successfully” matched pixels using a left-right consistency check [17]. Confidence measures are another example of evaluation in the absence of ground truth [11,32]. The idea is to measure the reliability of the calculated values for each pixel using heuristic or probabilistic approaches.

Approaches specifically designed for DAS have also been proposed. In [28,36], the authors proposed techniques that evaluate the generated stereo data if certain conditions are satisfied during the recording of the real-world input stereo-pairs. Recently, the organisers of the 2011 *DAGM* conference [6] provided its



**Fig. 2.** Sketch of the followed technique. The NCC index is calculated between the generated virtual image and the recorded control image.

own evaluation test-data set. However, there was no provided ground truth or an objective evaluation scheme for comparing results.

Generating ground truth for outdoor environments has also been investigated. For example, in [31] the true distance measurements were generated using a high-end *laser range-finder*. Despite the accuracy of the depth readings, the reduced resolution of the range finder, and possible calibration or synchronisation issues (between the cameras and the range finder), are still limiting the applicability of this option.

Extra images were used in [37] for defining a *prediction error* for *optic flow* and stereo-analysis (see also [34]). For our technique, we have adapted the prediction error technique for using it with three synchronous cameras recording uncontrolled environments. We use an evaluation index that takes under consideration the photometric differences between the three images involved in the analysis (which is a quite common situation in real-world environments).

### 3 Trinocular Evaluation Technique

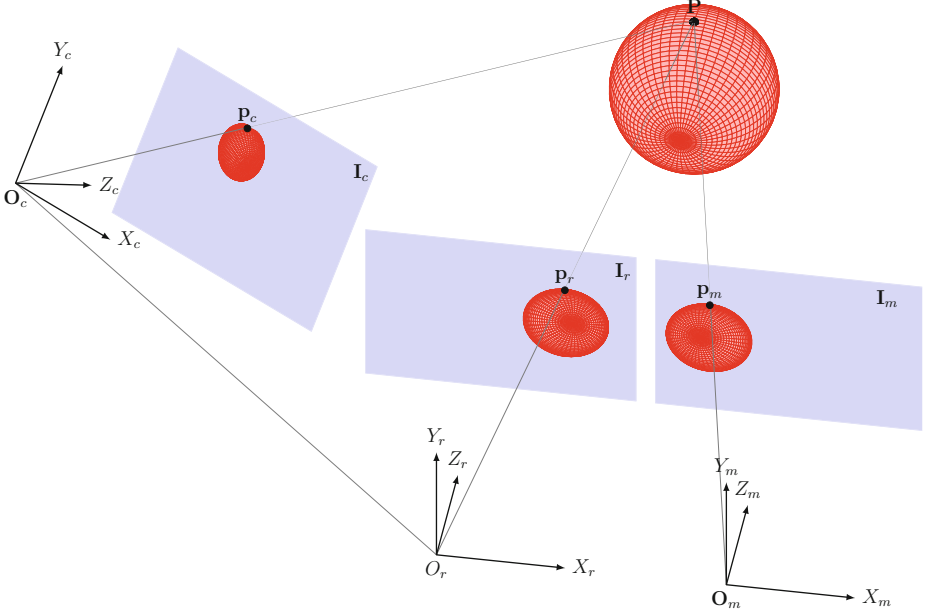
Consider time-synchronised recording of a scene by three video cameras. Video data captured by reference and match camera are rectified in such a way that each stereo-pair  $I_r$  and  $I_m$  satisfies the *standard stereo geometry* [21] (SSG). The third camera acts as control camera and is potentially in arbitrary pose “towards the scene recorded by reference and match camera”.

The objective is to generate a virtual image  $I_v$  from a disparity map calculated by a stereo-analysis algorithm (using  $I_r$  and  $I_m$ ), and to compare  $I_v$  with the control image  $I_c$  recorded with the control camera.

We generate  $I_v$  by mapping (warping) the pixels of the reference image  $I_r$  into the locations where they would have been recorded in  $I_c$ . Then,  $I_c$  and  $I_v$  are compared using *normalised cross-correlation* (NCC) as a measure; see Section 3.3 for its specification. Figure 2 summarises the followed technique.

#### 3.1 Common Forward Equations

Assume that the coordinate system of the reference camera is identified with the world coordinate system. Image coordinates are defined by each camera



**Fig. 3.** A general trinocular camera configuration. The two cameras, represented by the coordinate systems on the right, are assumed to satisfy the standard stereo geometry. The third camera is rectified with respect to internal camera parameters only (i.e. thus representing ideal central projection).

individually. Locations of reference, match and control camera are sketched in Figure 3. In world coordinates, the optical centre of the reference camera lies at the origin  $O_r = (0, 0, 0)^T$ , and those of the reference and match camera at  $O_m = (b, 0, 0)^T$  and  $O_c = (b_1, b_2, b_3)^T$ , respectively.

Let  $P = (X, Y, Z)^T$  be a scene point in the shared field of view of all the three cameras; and  $p_r = (x, y)^T \in I_r$ ,  $p_m = (x_m, y_m)^T \in I_m$ , and  $p_c = (x_c, y_c)^T \in I_c$  be the projections of  $P$  onto the rectified image planes of the three cameras. The corresponding image point in the virtual image is denoted by  $p_v = (x_v, y_v)^T$ .

For the assumed case of SSG between reference and match image, we provide a formula below to obtain the coordinates of  $p_v$  in terms of the coordinates of  $p_r$ , and the internal parameters of the stereo camera defined by the reference and match cameras (i.e., base-line distance  $b$  and unified focal length  $f$ ) and the corresponding disparity value  $d$  (computed by some stereo-vision algorithm) between  $p_r$  and  $p_m$ . Since  $P$  is visible from reference and match camera, by triangulation, it is possible to write the coordinates of  $P$  with respect to the coordinate system of  $I_r$  as follows:

$$(X, Y, Z)^T = \frac{b}{d}(x, y, f)^T \quad (1)$$

Now, let  $(X_c, Y_c, Z_c)^T$  be the coordinates of  $P$  with respect to  $O_c$ . Using homogeneous coordinates and (for abbreviation) letting  $\mathbf{C}$  and  $\mathbf{S}$  be short for *cosine* and *sine* functions, respectively, the matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{C}\gamma\mathbf{C}\beta & -\mathbf{C}\gamma\mathbf{S}\beta\mathbf{S}\alpha - \mathbf{S}\gamma\mathbf{C}\alpha & \mathbf{S}\gamma\mathbf{S}\alpha - \mathbf{C}\gamma\mathbf{S}\beta\mathbf{C}\alpha & -u_1 \\ \mathbf{S}\gamma\mathbf{C}\beta & \mathbf{C}\gamma\mathbf{C}\alpha - \mathbf{S}\gamma\mathbf{S}\beta\mathbf{S}\alpha & -\mathbf{S}\gamma\mathbf{S}\beta\mathbf{C}\alpha - \mathbf{C}\gamma\mathbf{S}\alpha & -u_2 \\ \mathbf{S}\beta & \mathbf{C}\beta\mathbf{S}\alpha & \mathbf{C}\beta\mathbf{C}\alpha & -u_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

specifies the mapping

$$(X_T, Y_T, Z_T, 1)^T = \mathbf{M} \cdot (X, Y, Z, 1)^T \quad (3)$$

where angles  $\alpha$ ,  $\beta$ , and  $\gamma$  represent a rotation that fixes the  $X, Y$  and  $Z$  axis, respectively.

$$u_1 = b_1\mathbf{C}\gamma\mathbf{C}\beta + b_2(-\mathbf{C}\gamma\mathbf{S}\beta\mathbf{S}\alpha - \mathbf{S}\gamma\mathbf{C}\alpha) + b_3(\mathbf{S}\gamma\mathbf{S}\alpha - \mathbf{C}\gamma\mathbf{S}\beta\mathbf{C}\alpha) \quad (4)$$

$$u_2 = b_1\mathbf{S}\gamma\mathbf{C}\beta + b_2(\mathbf{C}\gamma\mathbf{C}\alpha - \mathbf{S}\gamma\mathbf{S}\beta\mathbf{S}\alpha) + b_3(-\mathbf{S}\gamma\mathbf{S}\beta\mathbf{C}\alpha - \mathbf{C}\gamma\mathbf{S}\alpha) \quad (5)$$

$$u_3 = b_1\mathbf{S}\beta + b_2\mathbf{C}\beta\mathbf{S}\alpha + b_3\mathbf{C}\beta\mathbf{C}\alpha \quad (6)$$

Let  $m_{ij}$  be the element at position  $(i, j)$  in matrix  $\mathbf{M}$ , for  $1 \leq i, j \leq 3$  and  $f_c$  be the focal length of the control camera. Thus, using the equations defined by central projection, we have that

$$x_v = f_c \cdot \frac{m_{11}(bx - db_1) + m_{12}(by - db_2) + m_{13}(bf - db_3)}{m_{31}(bx - db_1) + m_{32}(by - db_2) + m_{33}(bf - db_3)} \quad (7)$$

$$y_v = f_c \cdot \frac{m_{21}(bx - db_1) + m_{22}(by - db_2) + m_{23}(bf - db_3)}{m_{31}(bx - db_1) + m_{32}(by - db_2) + m_{33}(bf - db_3)} \quad (8)$$

where  $d$  and  $b$  were defined above as being the disparity between pixels  $p_r$  and  $p_m$  and the length of the baseline between reference and match camera, respectively.

With these two *forward equations* [22] it is possible to map any pixel location  $(x, y)^T$  in the reference image into a pixel  $(x_v, y_v)^T$  in the image plane of the third camera. We select the nearest pixel position in this virtual image (i.e. in the pose of the third camera) because we do not aim at any visual improvement of this mapping (e.g. by interpolation of pixel values).

### 3.2 Poses of the Third Camera

In this section we discuss possible poses of the control camera. Note that the pose of the control camera defines the final appearance of the generated virtual image. The three cameras can be in an arbitrary position, but constrained by the fact that reference and match images need to satisfy SSG after rectification. In the following we denote the reference camera also as being the *left camera* of this pair of two rectified cameras.

In order to reduce the number of occluded points between reference and control camera, we aim at having the focal point of the control camera collinear



**Fig. 4.** Different types of occlusions for a horizontal configuration. *Left*: white pixels indicate occlusion between reference and match camera. *Centre*: black pixels indicate occlusion between third and reference camera (here: third camera is at position of match camera). *Right*: combined visualisation where third camera is now left of the reference camera.

with the focal points of the two other cameras. We discuss possible poses of the control camera in such a *horizontal configuration*.

Occluded points may cause areas with no texture in  $I_v$ , or pixels from  $I_r$  being mapped onto the wrong position due to having erroneous disparity results for occluded pixels in the stereo-pair. We illustrate this by examples generated using available ground truth for the synthetic sequence No. 1 from Set 2 of [7].

By increasing the translational distance between the poses of the control and the stereo-camera system, more occluded areas occur on  $I_v$ . Occlusions could be reduced (in general) by having the control camera positioned between reference and match camera. Figure 4 shows three different occlusion cases. For this figure we vary the poses of an imaginary third camera with respect to the used poses of reference and match camera when rendering this sequence. The disparity map  $I_d$  is the available ground truth.

On the left, the figure shows the virtual view corresponding to the pose of the reference image (i.e. the third camera was assumed to coincide with the reference camera). White pixels represent occluded pixels between reference and match image. Obviously, no disparity information is available for those. They are already occluded with respect to both stereo cameras. For the centre image of the figure, the third camera moved into the pose of the match camera. Occlusions are now shown in black, and correspond to occluded pixels between reference and control camera. The virtual view generated for a pose to the right of the reference image (in a horizontal configuration) would tend to “cover” also such occluded pixels that are visible for the reference camera but not for the match camera. On the right, the figure shows a virtual view based on the pose of the third camera located to the left of the reference camera. It is an example of a virtual view in which both kinds of occlusions occur (white and black).

For the first configuration there are no occlusions between reference and control camera. This configuration is actually known in self-consistency studies [27]. However, we are interested into using an additional image for the evaluation, not yet involved in the given stereo analysis, thus allowing us to obtain additional insights into the performance.

A symmetric pose of the control camera (focal point half-way on baseline between reference and match camera, with the tree optical axis parallel to each other) is also expected to minimise the impacts of both types of occlusions (i.e., the total number of either black or white pixels). In performance evaluation, it would be ideal to separate the impact of occlusions from those of incorrect stereo matching. Thus, the symmetric case seems to offer the possibility to focus on disparity errors. However, errors due to mismatches are actually often not as “obvious” for the symmetric case compared to a third-camera pose which differs (much) from the symmetric case.

Thus, altogether, an in-depth statistics about error distributions for different third-camera poses in a horizontal configuration (e.g. depending on scene geometry) might be of interest. However, in our practical tests we realised quickly that having the third camera in a “different pose” compared to the stereo-camera pair, but still “reasonably close” to this pair for not having too many occlusion issues, provides a better “challenge” than having a symmetric camera set-up.

The experiments reported in this chapter had the control camera approximately 50 cm to the left of the reference camera; reference and match camera are about 30 cm apart. This translational distance between control and reference camera appeared to be large enough for detecting miscalculated disparity values (even if disparities are small), but is still not yet exaggerating the influence of occluded points. Note that detecting the occluded regions could only be done by segmenting them manually on each trinocular stereo set. Thus, in the evaluation we only discard the obvious occlusions (i.e. those in the lateral border of the images).

### 3.3 Evaluation Index

As evaluation index we calculate the normalised cross-correlation (NCC) index between the virtual image  $I_v$  and the control image  $I_c$ , for each trinocular stereo frame at time  $t$  in a given image sequence. The NCC index is given by

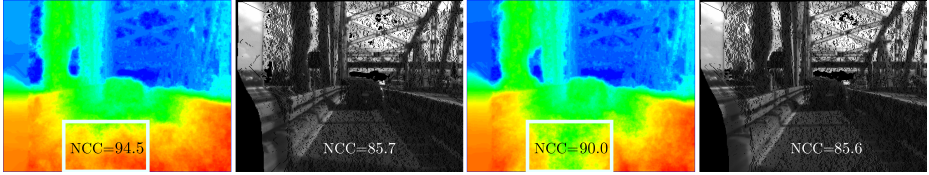
$$\text{NCC}(I_c, I_v) = \frac{100}{|\Omega|} \sum_{(x,y) \in \Omega} \frac{[I_c(x,y) - \mu_c][I_v(x,y) - \mu_v]}{\sigma_c \sigma_v} \quad (9)$$

where  $\mu_c$  and  $\mu_v$  denote the means, and  $\sigma_c$  and  $\sigma_v$  the standard deviations of the control and virtual images, respectively.

The set  $\Omega$  is a subset of all pixel locations. It needs to be selected for defining a “meaningful measure”. The default is that  $\Omega$  is simply defined by pixels having a valid disparity.<sup>2</sup>  $|\Omega|$  denotes the cardinality of this set.

The NCC index appears to be convenient for the presented evaluation technique (rather than, e.g., just a sum of absolute intensity differences), as it handles photometric differences between reference and control image to some degree, and brightness variations (e.g. non-uniform in a recorded image) are actually very typical for recorded outdoor videos.

<sup>2</sup> Our stereo-analysis algorithms assign a non-positive value to any pixel having no valid disparity.



**Fig. 5.** Samples of disparity maps and corresponding virtual images from consecutive frames from the *barrier sequence*. Both disparity maps show difference in disparity values in the indicated rectangular region, but the corresponding regions in the virtual images look almost the same. Thus, the NCC measure is expected to lead to about the same value (depicted in the disparity maps) in those regions. The NCC index for the full image is shown in the virtual images.

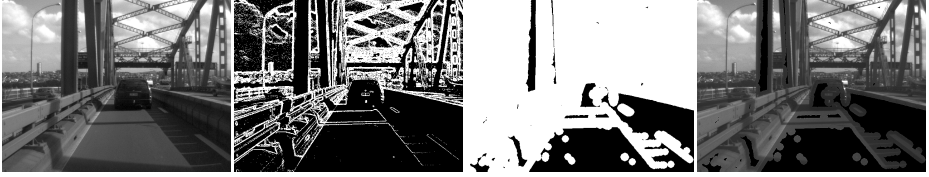
### 3.4 Alternative Approach for Defining Set $\Omega$

Images recorded in the context of DAS typically contain two large nearly textureless areas (i.e., featureless regions), namely the sky and the road. State-of-the-art stereo-analysis algorithms often have difficulties calculating correct disparities in such uniform regions. Values for the sky are irrelevant, and invalid values on the road (if properly detected) can be interpolated for identifying the road manifold.

We notice that the defined evaluation technique might report a good performance in such homogeneous regions even if this is not the case. In such regions it is very likely to occur that a pixel in the reference image with a corresponding miscalculated disparity value is mapped into a pixel in the virtual image that is in the same textureless region (i.e., a region with insignificant intensity differences between its pixels). Thus, values in this region may incorrectly influence the final evaluation index.

Figure 5 shows two virtual images and corresponding disparity maps when using the BPM-CEN stereo matcher (defined later in Section 4) for two consecutive stereo frames (frames 326 and 327 in the *barrier sequence*). A rectangular region is selected in the middle of the road; it shows differences in miscalculated disparities in both frames. However, the corresponding regions in the virtual images appear to be almost identical. For frame 326, disparity values in the rectangle are between 28 to 56, and between 21 to 41 for frame 327. For road surface points, this implies an average distance difference of about 5 metres. The evaluation index, restricted to the rectangle, does not show this defect, and it is considerable high for both frames (of 94.5 for frame 326 and of 90.0 for frame 327) compared to the NCC index calculated for the whole image (of 85.7 for frame 326 and of 85.6 for frame 327).

The following modified definition of set  $\Omega$  aims at restricting the performance evaluation to areas being “rich in texture”. The basic idea is as follows. Miscalculated disparities at, or within a small distance to pixels with a significant *intensity gradient* (used as a simple texture criterion) should affect the NCC index more than miscalculated disparities in textureless regions. One option is to



**Fig. 6.** Illustration of mask generation. From *left to right*: original image  $I$ , gradient image  $\nabla I$ , distance mask  $I_e$ , and identified textured zones in  $I$ .

simply discard the homogeneous regions completely when calculating the NCC index.

Given an image  $I$ , we generate a mask  $I_k$  that shrinks the domain  $\Omega$  by eliminating textureless regions. The image  $I_k$  is produced in three steps. First, a *binarized gradient image*  $\nabla I$  is defined as

$$\nabla I(x, y) = \begin{cases} 0, & \text{if } |(\partial_x I(x, y), \partial_y I(x, y))|_2 > T_1 \\ 1, & \text{otherwise} \end{cases} \quad (10)$$

where  $\partial_x$  (or  $\partial_y$ ) denote the partial derivative in the lateral (or vertical) direction.<sup>3</sup> The sign  $|\cdot|_2$  denotes the  $L_2$ -norm and  $T_1$  is an adjustable threshold. With  $\nabla I$  we aim to identify regions with some changes in intensity values.

The second step uses Euclidean distance transformation for generating an image  $I_e$  that labels pixels by their  $L_2$ -distance to edge pixels identified by  $\nabla I$ . Finally, we define  $I_k$  as

$$I_k(x, y) = \begin{cases} 0, & \text{if } I_e(x, y) > T_2 \\ 1, & \text{otherwise} \end{cases} \quad (11)$$

where  $T_2$  is again a predefined threshold. For the experiments reported in this chapter, we use the control image to define  $I_k$ , with  $T_1 = 5$  and  $T_2 = 10$ .

Figure 6 illustrates the process of generating  $I_k$ . The leftmost image is the input image  $I$ . The next image shows  $\nabla I$ ; followed by the distance image  $I_e$ . The resultant image  $I_k$  is shown in the rightmost position.

Alternatively, the distance values in  $I_e$  could be used as weights when defining the NCC index. However, experiments showed that using the defined mask  $I_k$  helps to calculate NCC indices which correspond, in general, with subjective visual evaluations of calculated depth accuracies.

## 4 Tested Stereo-Analysis Algorithms

We are interested in stereo-analysis algorithms for outdoor scenes in the context of DAS and related applications. The diversity of recording situations (e.g. in the night, in rain, with lighting artefacts) basically implies that one particular

<sup>3</sup> We use central differences to approximate the partial derivatives.



**Table 1.** Parametrization of the used stereo-analysis algorithms. BPM (SGM) uses identical values for number of iterations ( $c_1$ ) and level of tree ( $c_2$ ) for the three different cost functions SAD (sum of absolute differences), CEN (census function), or EPE (end-point error).

	BPM					GCM				SGM	
	dMax	sMax	$\lambda_d$	iteration	level	$\lambda_1$	$\lambda_2$	threshold	K	$c_1$	$c_2$
SAD	100	500	0.3			4.2	1.4	1	7		
CEN	75	600	0.6	7	6	3	1	1	5	30	150
EPE	33	200	0.225			2.6	0.86	16	4.33		

algorithm or parametrization cannot be the all-time winner; and some kind of adaptation (e.g. using different parametrization for the used matcher) needs to be supported.

#### 4.1 Three Matchers

For the experiments to be reported in this chapter, we selected three dense stereo-analysis algorithms based on algorithms that showed a good performance in previous studies [30,31]. We test them using three different cost functions and 8 long trinocular sequences. The parametrization used for each of the matchers (see Table 1), was optimised using the synthetic sequence introduced on Section 3.2.

*Belief-Propagation Matching* (BPM): We use a max-product iterative *belief propagation* algorithm as presented on [9]. This algorithm uses a truncation parameter for both, the cost function and the smoothness term. The smoothness term is a truncated quadratic function, which allows to obtain a smooth disparity map but without penalising depth discontinuities too much. Message passing is based on 4-adjacency. The original source code on [9] was modified to allow the use of different cost functions and of 10-bit input images as stereo frames.

To speed up the matching process, a hierarchical algorithm (i.e. a coarse-to-fine approach) is considered such that the passing of messages is more efficient when staying with a reduced number of iterations. The truncation parameters for the data (dMax) and the smoothness (sMax) terms, the weighting factor for the data term ( $\lambda_d$ ), the number of iterations (iteration), and number of levels (level) of the followed hierarchical algorithm are shown in Table 1.

*Graph-Cut Matching* (GCM): We use a modification of the *graph cut*-based algorithm presented in [25]. For minimising the energy function, a randomly initialised disparity map is considered as a weighted graph. The optimum disparity map is then calculated using the  $\alpha$ -*expansion method*. The implementation of this algorithm uses as smoothness term the binary Potts model to assure that a global minimum can be reached. The three parameters required for defining the Potts model ( $\lambda_1$ ,  $\lambda_2$ , and the threshold) and the weighting factor for the cost function ( $K$ ) are summarised on Table 1.

As for BPM, this algorithm was also modified such that a wider range of cost functions could be used.

*Semi-Global Matching* (SGM): We also use a semi-global matching algorithm as introduced in [17]. The matching strategy followed by BPM or GCM can be characterised as being potentially *global* (but practically limited by the number of iterations). In contrast, SGM limits its search space to a predefined set of *paths* to obtain an optimum disparity value only with respect to this selected search space. The used SGM implementation has been reported in [15]; it provides two SGM configurations that use the census transform as cost function (see Section 4.2), and optimise either along four or eight paths. We also examine the performance of the hierarchical SGM algorithm recently presented in [16]. Using four instead of eight paths reduces computation time without affecting much the quality of the disparity maps in general. The hierarchical algorithm increases the quality of SGM matching in areas that are the “usual suspects” for being complicated (e.g. non-textured areas such as on a road). The selected values for the two fixed penalties for the smoothness term ( $c_1$  and  $c_2$ ) are summarised in Table 1.

## 4.2 Three Cost Functions

Three cost functions are considered for our experiments. Each of them analyses different “characteristics” of the stereo input images when calculating costs for assigning a disparity value to a given pixel.

*Census Transform* (CEN): The census transform [41] is defined by the *Hamming* distance between two *signature vectors*. Its use supports robustness of a stereo matcher against common types of noise found in real-world images [15]. Following the latter paper, we use a  $9 \times 3$  neighbourhood as it favours a stronger data contribution along the epipolar line.

*Gradient-Based Cost Function* (EPE): The selected gradient-based cost function [20] analyses the  $L_1$ -distance between the end-points of the gradient vectors. This distance is expected to have a good performance when using real-world data [14]. To calculate the discrete partial derivatives that define the gradient vector, again we use central differences.

*Sum of Absolute Differences* (SAD): The sum-of-absolute-differences (SAD) cost function is an intensity-based similarity measure. It is known for having a poor performance when it comes to real-world stereo sequences, as the photometric consistency assumption is commonly violated in those data. We are interested in reconsidering this commonly used statement.

## 5 Experiments

We evaluate the performance of the three selected stereo-matchers using the three specified cost functions for BPM and GCM; the three presented configurations for SGM use CEN as a cost function. We use the abbreviations BPM-\* or GCM-\*, where \* denotes CEN, EPE, or SAD, and SGM-4, SGM-8, and SGM-HIER for the configurations of the semi-global matcher.

## 5.1 Evaluation Domains

The *full approach* refers to the method introduced at the beginning of Section 3; the *masked approach* denotes the method discussed in Section 3.4.

BPM and GCM algorithms generate usually a valid disparity (no matter whether correct or incorrect) for almost every pixel in the reference image. Thus, we compare the whole virtual and control image (except for the obviously occluded regions at the left margin of both images). As we are using the same evaluation domain, it is fair to compare the evaluation indices of those two algorithms (i.e. the *boosting effect* from the non-textured regions described in Section 3.4 should affect equally to indices of both algorithms).

For SGM, the evaluation domain is defined by the pixels in the disparity map detected as being valid (usually around 60% of the whole image domain). Thus, we only compare results between the three SGM configurations as their disparity maps have a similar amount of valid pixels.

## 5.2 Data Sets

Regarding the experimental data set, we use eight long (400 trinocular frames each) sequences recorded on real-world environments with test vehicle *HAKA1* (see 23), thus 9,600 test images in total, each of  $640 \times 480$  resolution at 10-bit per pixel.

The three cameras (of the same brand and model with identical lenses) were firmly mounted on an horizontal metal bar behind the windshield, just below the rear-view mirror. The reference and match cameras were placed on the driver's side of the vehicle. The length of the baseline is about 30 cm, thus, we are able to calculate distances to objects located from just less than 5 m to the cameras, up to around 310 m away (i.e., for a disparity value of 1). The control camera was fixed to the left of the rear-view mirror, at around 50 cm away from the reference camera. With this set-up we tried to keep the common field of view as large as possible. By keeping a considerable distance between reference and control cameras we support that appearing errors become more evident in the calculated NCC evaluation indices along a test sequence.

Four of our sequences were recorded on the same street (*the reference street*) under different environmental conditions. The street is surrounded by trees such that *illumination artefacts* 40 are present in the images, especially if the sun is low on the horizon. There are also some thin structures around the road (e.g., poles, trees branches, road signs) still make it a challenging test sequence. The surface of the road has actually sufficient texture so it is expected that the road will not be a source of noise during the matching process.

The other four sequences were recorded in more dynamic environments. They were recorded on busy roads, where moving pedestrians and vehicles are part of the scenery. Two of the sequences were recorded while driving at about 80 km/h, to test algorithms also for highly dynamic environments. The sequences are available for download in Set 9 from 7. A brief introduction to the sequences is as follows:



**Fig. 7.** Sample frames from the used 400-frame long trinocular sequences. *Top row*, from left to right: *midday sequence*, *wipers sequence*, *dusk sequence*, and *night sequence*. *Bottom row*, from left to right: *queen sequence*, *people sequence*, *harbour sequence*, and *barriers sequence*.

**Midday:** This sequence was recorded in the reference street under “ideal” conditions. The sun was close to its zenith, so there are not many of the undesired illumination artefacts. There is no incoming or oncoming traffic. The idea of recording such a simple sequence is to have a reference sequence, where the algorithms should perform best.

**Wiper:** In order to gain experience on the influence of varying occlusions of some regions in one (or both) camera(s) of the stereo system, we recorded a sequence while the wipers have been switched on (but no rain). This sequence was recorded within just a few minutes past the midday sequence on the same default road, expecting that the only “differing” factor for the matching process is the moving wipers.

**Dusk:** This sequence was recorded while having the sun in a position close to the horizon. The idea was to try to simulate the very common situation of having large saturated areas in one or in both cameras. As the road is surrounded by trees, there are intervals in the sequence with or without the sun striking directly into the cameras.

**Night:** This sequence was recorded at night. Almost all the light in the scene is provided by the headlamps of *HAKA1*. The trees around the road covered almost all the light from the lamp posts, which are very sparse in this particular road. The intention of having such a dark night scene was to simulate driving conditions as faced on second-order highways or rural roads.

**Queen:** This sequence was recorded on a main road of Auckland city. It has both, moving and static cars and pedestrians. It was recorded while driving towards a set of traffic lights, with a stop there. There are moving pedestrians at different distances. A bus stopped on the right hand side and has interesting reflections in its windows.

**People:** This sequence was recorded while *HAKA1* was standing still in front of a pedestrian crossing. The sequence has varying numbers of pedestrians in the

**Table 2.** Mean values of NCC indices, rounded to nearest integer, for full analysis. For each sequence, we highlight the best performing configuration for each algorithm.

		Full NCC average							
		Barriers	Dusk	Harbour	Midday	Night	People	Queen	Wiper
BPM	CEN	62	74	63	73	41	61	66	69
	EPE	<b>66</b>	<b>90</b>	<b>70</b>	<b>91</b>	<b>64</b>	<b>68</b>	<b>80</b>	<b>87</b>
	SAD	56	87	59	91	63	66	79	86
GCM	CEN	<b>59</b>	<b>87</b>	<b>62</b>	<b>93</b>	41	<b>66</b>	<b>82</b>	<b>89</b>
	EPE	37	82	38	88	21	42	67	83
	SAD	40	82	40	60	<b>62</b>	62	78	85
SGM	4	76	92	80	95	86	79	88	92
	8	76	92	80	95	87	79	<b>89</b>	92
	HIER	<b>76</b>	<b>95</b>	<b>81</b>	<b>96</b>	<b>90</b>	<b>79</b>	89	<b>94</b>

scene, between 1 up to around 20 at a time. The pedestrians walk only in two directions.

**Harbour:** This sequence was recorded while driving across the Auckland’s harbour bridge. The metal structure (i.e. the scaffolding) of the bridge represent a challenging collection of thin objects in different orientations. The shadows projected by the metal bars introduce interesting illuminations artefacts into the recorded images.

**Barriers:** This sequence was also recorded while driving across this harbour bridge. In this case the recording vehicle was driving in a lane that it is enclosed by medium-height concrete bars, and also the metal structure of the bridge is further up.

### 5.3 Results and Discussion

The discussion is focused on the most remarkable details of obtained results (e.g., when severe changes in the NCC index were detected, or when results between algorithms were particularly different). The average NCC indices for full or masked approaches, for all sequences and configurations, are presented in Tables 2 or 3, respectively.

**Midday:** All the algorithms performed “fairly well” (as expected); the indices reported for this sequence were the highest among the used sequences. Interestingly, all the algorithms had local minima at about the same trinocular stereo sets (see the left image on Figure 8). The drops in the indices are mainly caused by miscalculated disparities corresponding to thin structures (e.g. power poles or road signs). Erroneous disparity values on such objects had a particularly bad effect on NCC indices at those frames. This became even more obvious by using the masked analysis.

**Table 3.** Mean values of NCC indices, rounded to nearest integer, for masked analysis. For each sequence, we highlight the best performing configuration for each algorithm.

		Masked NCC average							
		Barriers	Dusk	Harbour	Midday	Night	People	Queen	Wiper
BPM	CEN	57	66	56	69	42	59	65	63
	EPE	<b>61</b>	<b>75</b>	<b>64</b>	<b>79</b>	<b>64</b>	<b>66</b>	<b>75</b>	<b>73</b>
	SAD	51	69	53	77	64	64	73	70
GCM	CEN	<b>54</b>	<b>76</b>	<b>54</b>	<b>82</b>	43	<b>66</b>	<b>78</b>	<b>77</b>
	EPE	33	64	32	67	28	39	57	60
	SAD	35	66	33	50	<b>63</b>	61	71	67
SGM	4	31	26	19	33	<b>34</b>	51	47	27
	8	<b>34</b>	33	<b>25</b>	38	31	<b>53</b>	<b>53</b>	33
	HIER	31	<b>41</b>	24	<b>45</b>	26	48	47	<b>49</b>

For GCM, the leading configuration was GCM-CEN. For the other two configurations, there are several regions with obvious (i.e. visual inspection) disparity miscalculations. They were correctly penalised with both evaluation analysis.

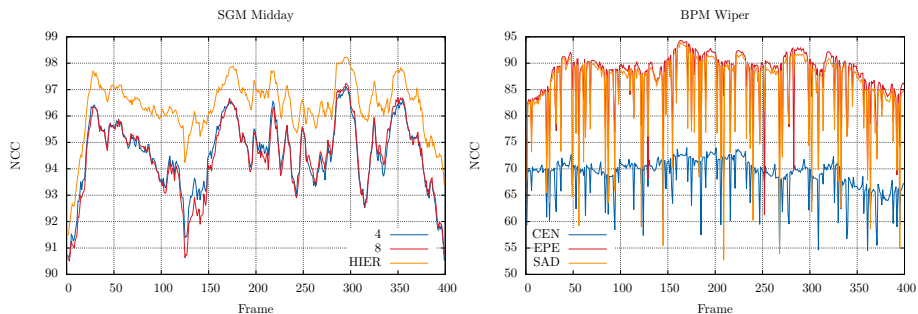
BPM-EPE and BPM-SAD reported the best (and very close to each other) NCC indices for BPM. For this matcher, the CEN cost function introduced a kind of a “salt-and-pepper” noise (i.e. non-homogeneous results were homogeneity is expected) into the disparity maps that was also clearly identified by our evaluation.

SGM-HIER showed a slightly better performance than the other two SGM configurations for full analysis. This could be due to a better performance of SGM-HIER on road regions (see the identified boosting effect of trinocular analysis). However, the same rank was observed when using the masked analysis, showing that the estimation in non-homogeneous regions had also been improved with the hierarchical algorithm. The left image in Figure 8 shows the full analysis’ results for the three SGM configurations. Note that there are almost no difference between SGM-4 and -8.

**Wiper:** This sequence represents a particular challenge for the trinocular analysis. The wipers might be occluded with respect to reference and match camera; or they might be occluded between the reference and control cameras. Thus, in this sequence, low NCC indices might not only be caused by miscalculated disparity values, but also due to having different objects (wipers) present in the virtual and the control image.

All the configurations showed a repetitive pattern of local minima, as expected. When the wipers were not present in any of the images, the algorithms performed just as with the midday sequence. Lowest local minima correspond to frames where the wipers were in the virtual image and not in the control one, or vice-versa; or when the wipers were in both images but in different position.

For cases when there were a wiper in the stereo-image pair, the algorithms handled the wipers as invalid pixels (SGM) or by propagating estimated disparity values of surrounding areas (BPM and GCM); this was more evident using CEN or EPE.



**Fig. 8.** *Left:* Midday sequence results for SGM. *Right:* Wiper sequence results for BPM.

Masked and full analysis led to similar results. With the masked approach, the local minima were not as low as with the full analysis. Miscalculations introduced by the wipers affected more the sky and road areas (we recall: both regions were ignored when using the masked approach).

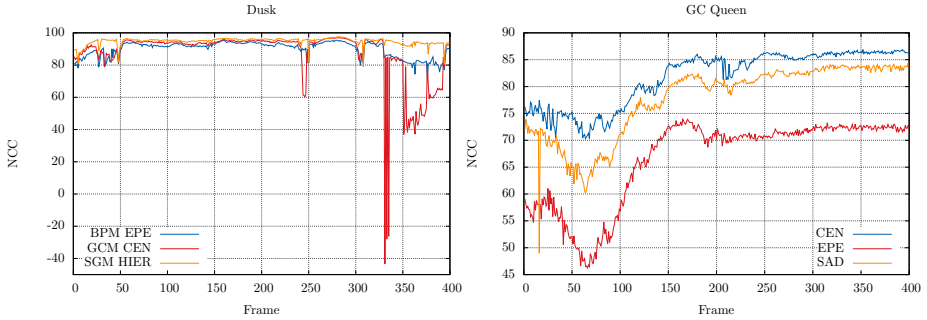
For BPM and GCM, the most negatively affected cost function was SAD; drops in magnitudes of indices were the largest compared to other cost functions. BPM-CEN was noticeably robust against the presence of wipers. Even that its average performance was worst than that of the other two BPM configurations (i.e. due to the detected salt-and-pepper noise), the drops in the index are not as large as for BPM-SAD or BPM-EPE. The left image on Figure 8 shows the results of the three BPM configurations. Note the large index’s drops for BPM-SAD. The repetitive pattern of local minima is similar for the configurations defined using the other two algorithms.

For SGM, -4 and -8 paths configuration showed an almost identical performance. The HIER configuration reported a more steadier performance (i.e. the drops on its index were smaller than those of the other two configurations); defining the best overall performing SGM configuration.

**Dusk:** As expected, the performance of all the configurations decreased when the sun stroke directly into the cameras (originating large homogeneous regions).

For all the algorithms and cost functions, there were scattered short time-intervals with an extreme low NCC index (e.g. around frame 250 or 300 on the right image on Figure 9). This was due to the fact that in those frames the sun struck only the control camera. Thus, there is an analogous effect as with the wiper sequence when there was a wiper only on the control camera. Ignoring those outliers, the shape of the plot increased or decreased depending on whether the sun struck directly into the three cameras, or not. For example, there were two time-intervals where the sun struck freely into the three cameras (say, [0,50] and [320,400]) and the evaluation reported lower indices than the average (see again Figure 9, left).

We stress the robustness of SGM (in particular for SGM-HIER) when comparing the three algorithms for this sequence. The indices of the three configurations are quite similar for most of the frames, but SGM-HIER kept a more stable performance in “complicated intervals” of the sequence. Figure 9, left, shows the



**Fig. 9.** *Left:* Dusk sequence results comparing best performing configurations of all three matchers. *Right:* Queen sequence results for GCM-configurations.

results for BPM-EPE, GCM-CEN, and SGM-HIER (the best performing configuration of each matcher on this sequence). Note that the low peaks are less intense for SGM. The rank suggested by this plot should be taken carefully, as image domains used for algorithm evaluation are different.

**Queen:** The results obtained with the full and masked analysis, for all the configurations, showed a common increasing tendency; which was particularly evident from frame 300. This is because the scene is less complex, only a few pedestrians remained on the field of view of the cameras and the road area had enough texture to be matched properly.

GCM-SAD and -EPE reported, between frames 20 and 100 (see Figure 9), a large decay in the indices when using the full approach. In this particular time-interval, the lower part of the road is quite homogeneous. Both configurations failed to match correctly this problematic region. The low peak was not reported by the masked approach, as this part of the road was discarded from the evaluation. None of the others configurations had particular problems with this region.

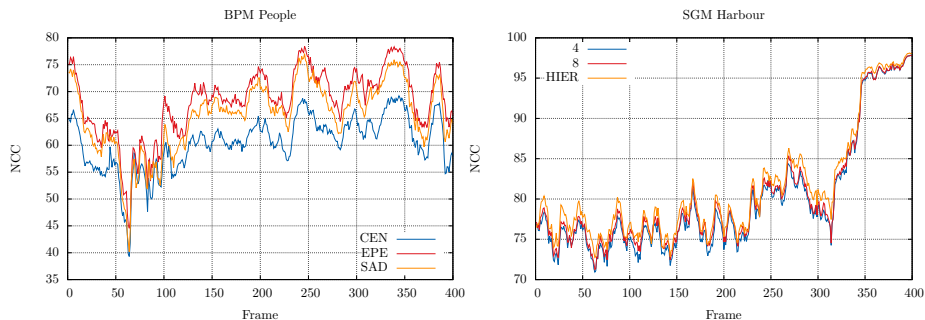
SGM-HIER and SGM-8 reported different rankings when using the full or the masked approach. It looks like SGM-HIER is taking advantage of the boosting effect from the full analysis for this sequence.

**People:** Results for all the algorithms showed a common pattern for masked and full analysis. Between frames 50 and 100, the two approaches reported low indices for all the configurations. This part of the sequence is the most busy one, with many pedestrians present in the scene. The following ups and downs correspond to a single (or two) pedestrian(s) entering or crossing the common field of view. See Figure 10, left, for BPM-results.

As the evaluation technique uses three different cameras, and all the pedestrians are fairly close to them, we might conclude that low indices (between frames 50 and 100) are due to occlusions between the cameras. But, as pedestrians are “fairly slim” structures, even a minor miscalculation implies a wrong reconstruction of the whole pedestrian in the virtual image (usually a misplaced body part).

The two evaluation approaches show an almost identical behaviour for BPM and GCM. For SGM, the ranks were totally different for the two types of analysis.





**Fig. 10.** *Left:* People sequence results for BPM. *Right:* Harbour sequence results for SGM.

It appears that SGM-HIER has more difficulties matching pixels near disparity discontinuities, but performs better on homogeneous regions.

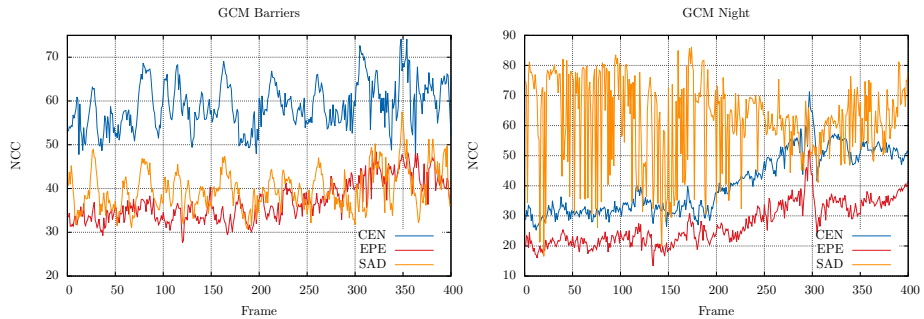
**Harbour:** This sequence reported an interesting difference between the full and the masked approach for SGM. In all the sequences analysed so far, no matter which algorithm, the masked analysis follows the same trend as the full analysis. For this sequence, for the three SGM-configurations, full or masked analysis reported a different behaviour in each case. In the full analysis, there was an increasing trend of the index along the sequence; this tendency of the index was particularly strong for the last 100 frames. The masked analysis reported an opposite tendency; the indices decrease along the sequence. In Figure 10, right, we show the results of the full analysis for SGM, where the increasing trend can easily be identified.

A possible explanation for this behaviour is that, at the end of the sequence, the metal structure of the bridge disappears from the scene. What is depicted in the images is mostly sky and road surface, with a large number of skinny poles and small buildings in the background. Increasing indices for the full analysis might be due to the boosting effect on the large homogeneous areas in the image. The decreasing tendency of the masked approach could be explained as even the smallest disparity miscalculation would imply a wrong warping of the skinny structures (i.e. the poles and buildings) in the scene. This irregular behaviour needs to be further analysed.

The masked and full approach for BPM and GCM both show the same pattern for all the three cost functions.

**Barriers:** For this sequence, the algorithms showed common trends, with differences in the magnitude of the index but still a common behaviour. The ups and downs in the indices (see Figure 11, left) were dictated by the appearance and disappearance of patches of the sky. The sky decreased indices and the metal structure of the bridge increased them; shadows created by the covering structure also contribute to increase the index as they made less homogeneous the road area.

The evaluation of the GCM results showed an interesting behaviour using the trinocular evaluation technique. The road and the barriers are large non-homogeneous areas that were well reconstructed in the virtual images generated



**Fig. 11.** *Left:* Barriers sequence results for GCM. *Right:* Night sequence result for GCM.

with GCM-SAD. Even though, the disparity maps are quite “blocky” (i.e. large regions were assigned with a single disparity value) in the road and barriers regions, while the upper part of the disparity images have plenty of miscalculated values. GCM-CEN generated smoother and less noisy disparity maps. We expected that, due to the boosting effect, the full analysis would assign really close NCC values for both configurations. However, it made a clear difference between the GCM-CEN and GCM-SAD results, assigning a larger NCC to the former configuration (see Figure 11, left). The difference between the evaluation indices was stressed using the masked approach.

**Night:** The matching of corresponding pixels, and the evaluation of the matching process using the trinocular technique are both challenging due to the limited dynamic range of the trinocular stereo sets from this sequence (of about 50 different intensity values only for some of the input images).

All the algorithms reported very low NCC indices. It is hard to visually identify the 3-dimensional structure of the scene in the disparity maps. The only exception were the SGM-HIER disparity images, where it is possible to identify the road area (illuminated by *HAKA1* headlamps) and even some of the objects that surround the road. This SGM-configuration reported the highest evaluation indices for the full analysis. However, it could not be identified as a better performer when using the masked analysis. The boosting effect of the correctly estimated road area seemed to help SGM-HIER in the full analysis.

The evaluation results for BPM and GCM show an increasing trend as the sequence reached the end. In the second half of the sequence there was more light available (an incoming vehicle with headlamps on is approaching, and the trees around the road are less dense), thus more disparity values were correctly calculated. Figure 11, right, depicts the results for the GCM configurations. Note the increasing tendency for the EPE and SAD configurations. The “volatile” indices reported in the first half of the sequence for GCM-SAD are due the assignation of a unique (i.e. but incorrect) very low disparity value to most of the pixels in the upper half of the frame. Due to the homogeneity of input images, the trinocular technique failed to assign a low NCC index (the masked analysis reported similar behaviour).

**Table 4.** Overall average NCC indices, rounded to nearest integer, for both approaches. The “Win” rows show for how many frames the specific configuration performed best. We compare BPM and GCM results directly, but separately from SGM, as image domains used for evaluation are different. For each approach, we highlight the best performing configuration for each algorithm; and that performing the best in a larger number of frames.

		BPM			GCM			SGM		
		CEN	EPE	SAD	CEN	EPE	SAD	4	8	HIER
Full	Avg	63	<b>77</b>	73	<b>72</b>	58	67	86	86	<b>88</b>
	Win	17	1449	82	<b>1479</b>	1	172	155	434	<b>2611</b>
Masked	Avg	60	<b>70</b>	65	<b>66</b>	55	45	33	38	<b>39</b>
	Win	24	1184	83	<b>1743</b>	0	166	353	<b>1434</b>	1413

#### 5.4 Overall Resume

Table 4 summarises the evaluation results for the two approaches. Each column represent the overall average for each one of the used configurations. Column “Win” (short for ‘winner’) shows the total number of frames on which a certain configuration outperformed all the others. Note that we compare the results of all the BPM and GCM configurations directly, but consider separately the results of SGM because the image domain  $\Omega$  used for BPM and GCM is different to that used for SGM.

Information obtained with the full and masked approach showed a good correlation. In the overall average, the two indices reported the same rankings. The masked approach was useful to stress miscalculations; as well as to discover well estimated disparities lost in regions full of miscalculated ones. But, we also noticed that the masked approach can also hide some miscalculated values. For example, in the queen sequence using GCM-SAD or -EPE, the mask analysis ignores completely a large miscalculated region of the road area that was correctly penalised with the full approach (i.e. the large decay in the index in the first 100 frames showed on Figure 9 left, is not reported on by the masked approach).

The results obtained in here suggest that the full approach, by itself, evaluates fairly and objectively the calculated disparity maps. However, both analysis should be considered when evaluating the algorithms.

The BPM algorithm showed an unexpected result when BPM-SAD outperformed BPM-CEN in the overall evaluation with respect to either evaluation approach (and for around 80 frames was the overall winner). The salt-and-pepper noise observed in the BPM-CEN disparity maps was severely penalised using both evaluations approaches. However, BPM-CEN had a more robust performance, its evaluation index is lower than that of BPM-SAD, but it showed a more steadier behaviour. For some problematic frames (e.g. in the dusk sequence), BPM-SAD generated “useless” data which was not the case for BPM-CEN.

Regarding GCM, the outperforming configuration was GCM-CEN. It generated noisy disparity measurements in homogeneous regions (i.e. the road), but managed to reconstruct better the other structures present in the scenes.

The salt-and-pepper noise observed in the BPM-CEN disparity maps, was not detected for this configuration. GCM-SAD estimated the homogeneous regions as single-valued blocks and introduced “a lot” of incorrect measurements everywhere else. The full approach was in general capable to fairly evaluate the blocky behaviour observed with this configuration; by using the masked approach the miscalculations become more evident. GCM-EPE had the poorest performance; the disparity maps have considerable amounts of random values, which degraded significantly the generated virtual images.

Among the SGM configurations, SGM-HIER showed the best overall performance with the two approaches. SGM-8 and -4 obtained very similar evaluation indices using the full approach. The results obtained using the masked approach suggest that using more paths improve the disparity computation in non-homogeneous regions (e.g. the difference in magnitude between the 4- and 8-paths configuration was larger with the masked approach, see Table 4).

The most noticeable difference between the three configurations was detected in the estimation of homogeneous regions. SGM-HIER generated more uniform surfaces but, it seems that its NCC-indices were “helped” by the boosting effect when using the full analysis. Note that for the masked approach, even that SGM-HIER had the best (masked) average index, SGM-8 performed better in a larger number of frames.

## 6 Conclusions

This chapter reported about an evaluation technique for stereo-analysis algorithms that uses an extra image (besides of the input stereo-pair) as reference data. We illustrated its efficacy by measuring the performance of three different algorithms using eight long real-world sequences. The discussed trinocular technique [or, say  $(n + 1)$ -ocular analysis for an  $n$ -camera stereo-vision system] appears to be a fairly indicative tool to highlight issues or good performance of the tested stereo-analysis methods. For designing an adaptive computer vision solution for vision-based driver assistance, it appears as particularly interesting to identify frames (or time intervals) where the behaviour of stereo-vision algorithms “suddenly changes”, such that a new optimisation can take place for selecting and configuring a suitable matcher.

Large homogeneous regions in the images might mislead the NCC evaluation index; we suggested an alternative method (the masked approach) to avoid those problematic regions. By using both the full and the masked approach, it was possible to point out particular weakness or strengths of a matching algorithm depending on the used configuration. Miscalculations in homogeneous areas may not become “visible” due to ongoing high NCC-indices in the full analysis; however, using the masked approach, a more appropriate evaluation is possible in general.

The proposed trinocular technique seems to be an adequate answer to the problem of finding an objective evaluation method in the absence of ground-truth. A (relatively) simple hardware set up allows us to record trinocular data sets as appropriate in particular stereo-vision applications.

## References

1. Badino, H., Franke, U., Mester, R.: Free space computation using stochastic occupancy grids and dynamic programming. In: Proc. Dynamic Vision, ICCV Workshop, pp. 1–12 (2007)
2. Banks, J., Corke, P.: Quantitative evaluation of matching methods and validity measures for stereo vision. *Int. J. Robotics Research* 20, 512–532 (2001)
3. Bolles, R., Baker, H., Hannah, M.: The JISCT stereo evaluation. In: ARPA Image Understanding Workshop, pp. 263–274 (1993)
4. CMU/VASC. Stereo image data base, <http://vasc.ri.cmu.edu/idb/html/stereo/> (retrieved 2012)
5. Computer Vision Group, University of Bonn. Stereo images with ground truth disparity and occlusion, [http://www.uni-bonn.de/uzs751/MRTStereo/stereo\\_data/index.html](http://www.uni-bonn.de/uzs751/MRTStereo/stereo_data/index.html) (retrieved 2012)
6. DAGM 2011, adverse vision condition challenge, <http://www.dagm2011.org/adverse-vision-conditions-challenge.html> (retrieved 2012)
7. The .*enpeda.* project, The University of Auckland. EISATS, Set 2, <http://www.mi.auckland.ac.nz/EISATS> (retrieved 2012)
8. Faugeras, O., Fua, P., Hotz, B., Ma, R., Robert, L., Thonnat, M., Zhang, Z.: Quantitative and qualitative comparison of some area and feature-based stereo-analysis algorithms. In: Proc. Workshop Robust Computer Vision, pp. 1–26 (1992)
9. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. *Int. J. Computer Vision* 70, 41–54 (2006)
10. Georgescu, B., Meer, P.: Point matching under large image deformations and illumination changes. *IEEE Trans. Pattern Anal. Mach. Intel.* 26, 674–688 (2004)
11. Gherardi, R.: Confidence-based cost modulation for stereo matching. In: Proc. ICPR (2008) 978-1-4244-2175-6
12. Gülch, E.: Results of test on image matching of ISPRS WG III/4. *ISPRS J. Photogrammetry Remote Sensing* 46, 1–18 (1991)
13. Haeusler, R., Klette, R.: Benchmarking Stereo Data (Not the Matching Algorithms). In: Gesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) DAGM 2010. LNCS, vol. 6376, pp. 383–392. Springer, Heidelberg (2010)
14. Hermann, S., Vaudrey, T.: The gradient - a powerful and robust cost function for stereo matching. In: Proc. IVCNZ (2010) 978-1-4244-9631-0
15. Hermann, S., Morales, S., Klette, R.: Half-resolution semi-global stereo matching. In: Proc. IEEE Symp. IV, pp. 201–206 (2011)
16. Hermann, S., Klette, R.: Evaluation of a New Coarse-to-Fine Strategy for Fast Semi-Global Stereo Matching. In: Ho, Y.-S. (ed.) PSIVT 2011, Part I. LNCS, vol. 7087, pp. 395–406. Springer, Heidelberg (2011)
17. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Proc. CVPR, vol. 2, pp. 807–814 (2005)
18. JISCT Stereo Images, <http://vasc.ri.cmu.edu/idb/html/jisct/index.html> (retrieved 2012)
19. Klappstein, J., Vaudrey, T., Rabe, C., Wedel, A., Klette, R.: Moving Object Segmentation using Optical Flow and Depth Information. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 611–623. Springer, Heidelberg (2009)

20. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Proc. ICPR (2006), 10.1109/ICPR.2006.1033
21. Klette, R., Schlüns, K., Koschan, A.: Computer vision: three-dimensional data from images. Springer, Singapore (1998)
22. Klette, R., Zamperoni, P.: Handbook of Image Processing Operators. John Wiley & Sons, Inc. (1996)
23. Klette, R., Krüger, N., Vaudrey, T., Pauwels, K., Hulle, M., Morales, S., Kandil, F., Haeusler, R., Pugeault, N., Rabe, C., Leppe, M.: Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database. IEEE Trans. Vehicular Technology 60, 2012–2026 (2011)
24. Kogler, J., Hemetsberger, H., Alefs, B., Kubinger, W., Travis, W.: Embedded stereo vision system for intelligent autonomous vehicles. In: Proc. IEEE Symp. IV, pp. 64–69 (2006)
25. Kolmogorov, V., Zabih, R.: Graph cut algorithms for binocular stereo with occlusions. In: Math. Models in Computer Vision: The Handbook, pp. 423–437. Springer (2005)
26. Kondermann, D., Meister, S., Lauer, P.: An outdoor stereo camera system for the generation of real-world benchmark datasets with ground truth. Universität Heidelberg HCI, Technical Rep. (2011)
27. Leclerc, Y.G., Luong, Q.-T., Fua, P.: Measuring the Self-Consistency of Stereo Algorithms. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 282–298. Springer, Heidelberg (2000)
28. Liu, Z., Klette, R.: Approximate Ground Truth for Stereo and Motion Analysis on Real-World Sequences. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 874–885. Springer, Heidelberg (2009)
29. Mohan, R., Medioni, G., Nevatia, R.: Stereo error detection, correction and evaluation. IEEE Trans. Pattern Analysis Machine Intelligence 11, 113–120 (1989)
30. Morales, S., Klette, R.: A Third Eye for Performance Evaluation in Stereo Sequence Analysis. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 1078–1086. Springer, Heidelberg (2009)
31. Morales, S., Klette, R.: Ground Truth Evaluation of Stereo Algorithms for Real World Applications. In: Koch, R., Huang, F. (eds.) ACCV 2010 Workshops, Part II. LNCS, vol. 6469, pp. 152–162. Springer, Heidelberg (2011)
32. Mordohai, P.: The self-aware matching measure for stereo. In: Proc. ICCV, pp. 1841–1848 (2009)
33. Satoh, Y., Sakaue, K.: An omnidirectional stereo vision-based smart wheelchair. EURASIP J. Image Video Processing, 1–12 (2007)
34. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Computer Vision 47, 7–42 (2001)
35. Schauwecker, K., Morales, S., Hermann, S., Klette, R.: A comparative study of stereo-matching algorithms for road-modelling in the presence of windscreen wipers. In: Proc. IEEE Symp. IV, pp. 7–12 (2011)
36. Steingrube, P., Gehrig, S.K., Franke, U.: Performance Evaluation of Stereo Algorithms for Automotive Applications. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 285–294. Springer, Heidelberg (2009)
37. Szeliski, R.: Prediction error as a quality metric for motion and stereo. In: Proc. ICCV, pp. 781–788 (1999)

38. van der Mark, W., Gavrilu, M.: Real-time dense stereo for intelligent vehicles. *IEEE Trans. Intelligent Transportation Systems* 7, 38–50 (2006)
39. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: *Proc. IVCNZ*, pp. 1–6 (2008)
40. Vaudrey, T., Morales, S., Wedel, A., Klette, R.: Generalized residual images effect on illumination artifact removal for correspondence algorithms. *Pattern Recognition* 44, 2034–2046 (2011)
41. Zabih, R., Woodfill, J.: Non-Parametric Local Transforms for Computing Visual Correspondence. In: Eklundh, J.-O. (ed.) *ECCV 1994, Part II*. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)

# Pyramid Transform and Scale-Space Analysis in Image Analysis

Yoshihiko Mochizuki<sup>1</sup> and Atsushi Imiya<sup>2</sup>

<sup>1</sup> Faculty of Science and Engineering, Waseda University, Japan  
Okubo 3-4-1, Shinjuku-ku, Tokyo 169-8555, Japan

<sup>2</sup> Institute of Media and Information Technology, Chiba University, Japan  
Yayoicho 1-33, Inage-ku, Chiba, 263-8522, Japan

**Abstract.** The pyramid transform compresses images while preserving global features such as edges and segments. The pyramid transform is efficiently used in optical flow computation starting from planar images captured by pinhole camera systems, since the propagation of features from coarse sampling to fine sampling allows the computation of both large displacements in low-resolution images sampled by a coarse grid and small displacements in high-resolution images sampled by a fine grid.

The image pyramid transform involves the resizing of an image by downsampling after convolution with the Gaussian kernel. Since the convolution with the Gaussian kernel for smoothing is derived as the solution of a linear diffusion equation, the pyramid transform is performed by applying a downsampling operation to the solution of the linear diffusion equation.

## 1 Introduction

The purpose of this paper is twofold. First, we introduce a method to construct the pyramid transform on curved manifolds. Second, we propose a method to evaluate the performance of optical flow without ground truth.

In images captured by an omnidirectional imaging system, moving objects and target objects are relatively sparse, since the system images a wide-view environment in a single view. The pyramid transform compresses a wide-view image to a small image while preserving the global features of the images. Therefore, pyramid transforms are suitable for the preprocessing of an omnidirectional image/image sequence. However, omnidirectional images are geometric images on a curved manifold. Therefore, we are required to construct the pyramid transform [1, 2] for the multiresolution representation [3] of images on a sphere.

The real-world images captured by an imaging system mounted on a car and on a mobile robot used for navigation and understanding of the environment have no ground truth. Therefore, for the evaluation of computer vision algorithms in a large real-world environment, we are required to compute features and evaluate the results simultaneously. For stereo reconstruction, performance evaluation without ground truth is achieved by using a parallel trinocular system, that is,



a pair of stereo images is used for the computation and the other pair of images is used for evaluation [4]. In featureless flow-based navigation, three consecutive images are used for the detection of free space for motion planning, that is, the optical flow field computed from the first pair of images is used to estimate the free space in the third image [5].

Since the pyramid transform reduces the size of images while preserving the size of pixels, the transform is used for preprocessing in the analysis of sparse images [6–11]. The pyramid-transform-based multiresolution method efficiently computes both large-displacement and small-displacement motion by propagating global features in a coarse grid to a fine grid [1, 12]. The pyramid-transform-based method is efficiently used for optical flow computation from usual images captured by pinhole camera systems. The image pyramid is separated into the smoothing operation by the convolution with Gaussian kernel [13–16] and the resizing operation on images by a downsampling operation [17, 18]. Since convolution with the Gaussian kernel for smoothing is achieved by computing the solution of a diffusion equation [19–22], the pyramid transform is achieved by the downsampling operation to the solution of the diffusion equation. We introduce the Gaussian pyramid transform using scale-space analysis on the sphere. Since the resolution of images captured by catoptrics and dipodic omnidirectional camera systems is nonuniform, the pyramid-based multiresolution analysis allows us to compute features on images uniformly using feature propagation across the resolutions. The Gaussian pyramid transform on the plane is performed by downsampling of the convolution between an image and a kernel function. Since the convolution with the Gaussian kernel is the solution of a linear diffusion equation, the Gaussian pyramid is obtained by applying downsampling to the solution of linear diffusion equation. Here, we extend this idea.

In a real environment, the payload of a mobile robot, for example, the power supply, the capacity of input devices and the computing power, is restricted. Therefore, mobile robots are required to have simple mechanisms and devices [5, 23] for navigation and localisation. To achieve a low payload, navigation algorithms for autonomous robots using a vision system inspired by insects have been proposed [24–28]. The insect-inspired vision system for robot control uses simple information observed by a vision system mounted on the robot. The view from the eyes of flying birds and the compound eyes of insects is a spherical image, which is a normalised image captured by an omnidirectional vision system.

There are two typical methods for optical flow estimation for pinhole images, the Lucas-Kanade (LK) method [29, 30] and the Horn-Schunck (HS) method [31], which are a template-matching-based method and a variational-based method, respectively. The image pyramid technique is commonly used to refine the accuracy of the stability of optical flow. The image pyramid is constructed by smoothing by Gaussian blurring and by resizing by downsampling. The LK method with pyramid-based multiresolution optical flow computation (LKP) [29, 30] is used to guarantee the accuracy and stability of the solution for the image sequence observed by a conventional pinhole camera. The convolution with the Gaussian kernel in the pyramid transform is computed using a discretized small kernel for

an planar image, for example, a  $5 \times 5$  window is a typical selection for the kernel assuming that the image is planar in this region.

A mathematical background of the multi-resolution image analysis based on the pyramid transform [12, 32, 33] is the algebraic multigrid method in numerical linear algebra [34–36]. The core of the algebraic multigrid method is the reduction of the equation to the coarse grid for accurate estimation of the residual and the expansion of the residual to the finer grid as a correction of solution. The expanded residual globally corrects the solution of the equation in the finer grid. The injection and full-weighting are fundamental reduction operations [34]. The full-weighting reduction is equivalent to the pyramid transform in digital image analysis. The pyramid transform is effectively used in optical flow computation [17, 30]. In optical flow computation with the pyramid transform, the correction procedure of the solution using the solution in the coarse grid is called the multiresolution warp. In this paper, we develop a full-weighting multiresolution linear-equation solver on the sphere [36] for the optical-flow computation from a spherical omni-directional image sequence. In meteorology, the multigrid method on the sphere is used to solve numerically partial differential equations on the sphere for the global weather prediction [37–39]. Since in our omnidirectional image analysis, a diffusion equation on the sphere is numerically solved for the computation of optical flow of spherical image sequence.

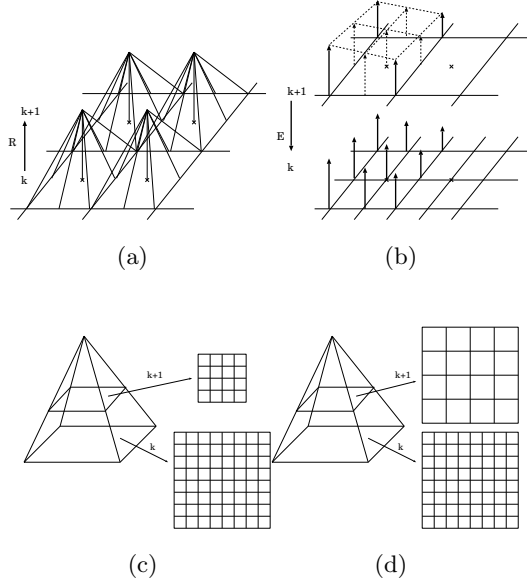
## 2 Mathematical Aspects of the Pyramid Transform

**Reduction and Expansion.** For the sampled function  $f_{ij} = f(i, j)$ , the pyramid transform  $R$  [6–11] and its dual transform  $E$  [12, 17, 18] are expressed as

$$Rf_{mn} = \sum_{i,j=-1}^1 w_i w_j f_{2m-i, 2n-j}, \quad Ef_{mn} = 4 \sum_{i,j=-2}^2 w_i w_j f_{\frac{m-i}{2}, \frac{n-j}{2}}, \quad (1)$$

where  $w_{\pm 1} = \frac{1}{4}$  and  $w_0 = \frac{1}{2}$ , and the summation is over integer values of  $\frac{(m-i)}{2}$  and  $\frac{(n-j)}{2}$ . Figures 1(a) and 1(b) show the transforms  $R$  and  $E$ , respectively. The operation  $R$  in each step is performed by computing a weighted average of the image values in a finite small region, which is called the window for the operation. Therefore, image features extracted in the higher-layer images of the pyramid transform describe global properties, in contrast with those extracted in the lower layers in the hierarchical expression, as shown in Figs. 1(c) and 1(d). Furthermore, the operation  $E$  is achieved by linear interpolation. These two operations involve the reduction and expansion of the image sizes.

Traditionally, the pyramid transform [6–12, 17, 18] yields a reduced image sequence. This interpretation is drawn from a fundamental property that the pyramid transform in each step yields a smaller image through a downsampling process if we use the same pixel size in each layer to express digital images. However, if we use the same landscape size in each layer, the transform in each step yields a lower-resolution image.



**Fig. 1.** Pyramid transform and its dual transform. (a) Reduction and (b) expansion for a 2D image. (c) Pyramid transform with equi-pixel size in each layer. (d) Pyramid transform with equi-image size in each layer.

A generalisation of the transforms defined in eq. (1) is

$$R^k f_{mn} = \sum_{i,j=-1}^1 w_i^k w_j^k f_{2^k m-i, 2^k n-j}, \quad E^k f_{mn} = 4^k \sum_{i,j=-2}^2 w_i^k w_j^k f_{\frac{m-i}{2^k}, \frac{n-j}{2^k}}, \quad (2)$$

where  $w_{\pm i}^k = \frac{1}{2^k} (1 - \frac{1}{2^k} |i|)$ ,  $|i| \leq 2^k$ .

Setting the matrix  $\mathbf{B}$  to be

$$\mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}, \quad (3)$$

the matrix expression of the Gaussian convolution kernel [41] is

$$\mathbf{G} = \frac{1}{4} (\mathbf{I} + \frac{1}{2} \mathbf{I} \otimes \mathbf{B} + \frac{1}{2} \mathbf{B} \otimes \mathbf{I} + \frac{1}{4} \mathbf{B} \otimes \mathbf{B}). \quad (4)$$

Furthermore, the downsampling operation [42] is expressed as

$$\mathbf{H} = (\mathbf{I} \otimes \mathbf{e}_2^\top) \otimes (\mathbf{e}_2^\top \otimes \mathbf{I}), \quad (5)$$

where  $\mathbf{e}_2 = (0, 1)^\top$ . The upsampling operation [42] is expressed as  $\mathbf{H}^\top$ . Since the Gaussian pyramid transform is performed by applying downsampling to the Gaussian convolution of the image array, the matrix forms of the Gaussian pyramid transform and the dual transform are  $\mathbf{R} = \mathbf{H}\mathbf{G}$  and  $\mathbf{E} = \mathbf{G}\mathbf{H}^\top$ , respectively. From these matrix expressions, we have the relation [4]  $(\mathbf{R}^k)^\top = ((\mathbf{H}\mathbf{G})^k)^\top = (\mathbf{G}\mathbf{H}^\top)^k = \mathbf{E}^k$ .

The spectra of  $\mathbf{E}^k$  and  $\mathbf{R}^k$  satisfy the relations  $\rho(\mathbf{E}^k) \leq 1$  and  $\rho(\mathbf{R}^k) \leq 1$  since  $\rho(\mathbf{G}) \leq 1$ ,  $\rho(\mathbf{D}) \leq 1$ ,  $\rho(\mathbf{H}^\top) = \rho(\mathbf{H}) \leq 1$ ,  $\rho(\mathbf{E}) \leq \rho(\mathbf{G})\rho(\mathbf{H}^\top)$  and  $\rho(\mathbf{R}) \leq \rho(\mathbf{H})\rho(\mathbf{G})$ .

**Scale Space Analysis and Pyramid Transform.** The pyramid transform

$$g_n := \frac{1}{4}f_{2n-1} + \frac{1}{2}f_{2n} + \frac{1}{4}f_{2n+1} = \frac{1}{4}(f_{2n-1} + 2f_{2n} + f_{2n+1}) \quad (6)$$

for the sequence  $\{f_n\}_{n=-\infty}^\infty$  is rewritten as

$$g_n = h_{2n}, \quad h_n = \frac{1}{4}(f_{n-1} + 2f_n + f_{n+1}). \quad (7)$$

These relations imply that the pyramid transform is performed by downsampling after calculating the moving average. If we adopt the discrete Gaussian  $k_i$  for the smoothing kernel, the pyramid transform is

$$g_n = \sum_{i=-\infty}^{\infty} k_{2n-i}f_i, \quad g_n = h_{2n}, \quad h_n = \sum_{i=-\infty}^{\infty} k_{n-i}f_i. \quad (8)$$

For an analog function, downsampling after convolution is expressed as

$$g(x) = h(\sigma x, \tau), \quad h(x, \tau) = \int_{-\infty}^{\infty} k_\tau(x-y)f(y)dy. \quad (9)$$

If

$$k_\tau(x) = \frac{1}{\sqrt{2\pi\tau}} \exp\left(-\frac{x^2}{2\tau}\right), \quad (10)$$

$h(x, \tau)$  is the solution of

$$\frac{\partial h}{\partial \tau} = \frac{1}{2} \frac{\partial^2 h}{\partial x^2} \quad (11)$$

for  $h(x, 0) = f(x)$ . Moreover, discretisation of the diffusion equation

$$h_i^{(n+1)} - h_i^{(n)} = \frac{1}{2} \left( \frac{h_{i+1}^{(n)} - 2h_i^{(n)} + h_{i-1}^{(n)}}{2} \right) \quad (12)$$

derives the discrete convolution

$$h_i := \frac{1}{4}h_{i+1} + \frac{1}{2}h_i + \frac{1}{4}h_{i-1}. \quad (13)$$

<sup>1</sup> Setting  $\mathbf{G}_k$  to be the matrix expression of the convolution in the first equation of eq. (2), we have the relation  $(\mathbf{H}\mathbf{G})^k = \mathbf{D}^k \mathbf{G}_k$ , where  $\mathbf{G}_k \neq \mathbf{G}^k$ .

From eqs. (9), (10), and (11) for two-dimensional functions, we have the relations

$$g(x, y) = h(\sigma x, \sigma y; \tau) \tag{14}$$

$$h(x, y; \tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_{\tau}(x - u, y - v) f(u, v) dudv, \tag{15}$$

$$k_{\tau}(x, y) = \frac{1}{2\pi\tau} \exp\left(-\frac{x^2 + y^2}{2\tau}\right), \tag{16}$$

since  $h(x, y, \tau)$  is the solution of

$$\frac{\partial}{\partial \tau} h = \frac{1}{4} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) h \tag{17}$$

for  $h(x, y, 0) = f(x, y)$ .

For a function <sup>2</sup>such that  $w_{\sigma}(x) = w_{\sigma}(-x) \geq 0$  and  $w_{\sigma}(x) = 0$  for  $|x| > \sigma$ , we deal with the linear transforms

$$g(x, y) = Rf(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w_{\sigma}(u)w_{\sigma}(v)f(\sigma x - u, \sigma y - v)dudv, \tag{18}$$

$$f(x, y) = Eg(x, y) = \frac{1}{\sigma^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w_{\sigma}(u)w_{\sigma}(v)g\left(\frac{x - u}{\sigma}, \frac{y - v}{\sigma}\right) dudv. \tag{19}$$

These transforms are shift-invariant downsampling and upsampling operations with orders  $\sigma$  and  $\sigma^{-1}$ , respectively. Therefore, the results of the operations  $Rf$  and  $Eg$  are elements of the Sobolev space.

We set

$$R^{k+1}f(x, y) = R(R^k f)(x, y), \quad E^{k+1}f(x, y) = E(E^k f)(x, y), \quad k \geq 1. \tag{20}$$

The sequence  $\{f_{(k)} = R^k f\}_{k=0}^K$  expresses a hierarchical expression for the image  $f$ .

**Definition 1.** *In both the defined domain and the range space of the transformation  $R$ , the inner products of functions are defined as*

$$(f, g)_D = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)g(x, y)dx dy, \tag{21}$$

$$(Rf, Rg)_R = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Rf(x, y)Rg(x, y)dx dy. \tag{22}$$

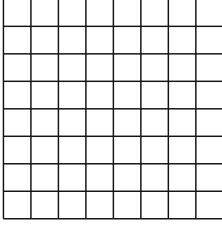
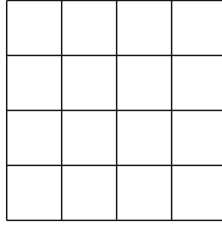
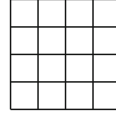
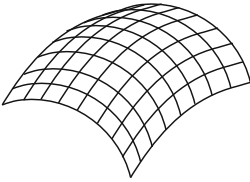
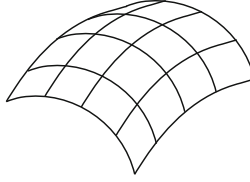
The dual operation  $R^*$  of the operation  $R$  satisfies the relation  $(f, Rg)_R = (R^*f, g)_D$ .

---

<sup>2</sup> For  $\sigma > 0$

$$w_{\sigma}(x) = \begin{cases} \frac{1}{\sigma} \left(1 - \frac{1}{\sigma}|x|\right) & |x| \leq \sigma \\ 0 & |x| > \sigma, \end{cases}$$

is a generalisation of  $w_{\pm} = \frac{1}{4}$  and  $e_0 = \frac{1}{2}$ .

(a) Fine grid  $\mathbf{D}$ (b) Coarse grid  $\mathbf{D}$ (c) Shrunk grid  $\mathbf{D}$ (d) Fine grid on  $\Gamma_D$ (e) Coarse grid on  $\Gamma_D$ (f) Shrunk grid on  $\Gamma_D$ 

**Fig. 2.** Pyramid transform on the manifold  $\Gamma$ . Using an appropriate bijection  $\xi = \phi(\mathbf{x})$  from  $\mathbf{D} \subset \mathbb{R}^2$  to  $\Gamma_D \subset \mathbb{R}^3$ , the downsampling operation is achieved in the subset  $\Gamma_D$  on the manifold  $\Gamma$ .

For the operators  $R$  and  $E$ , the relation

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Rf(x, y)g(x, y)dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)Eg(x, y)dx dy \quad (23)$$

is satisfied. Therefore, we have the relation  $R^* = E$ . For the derivation. Furthermore, eqs. (20) and (23) imply that the dual operation of  $R^k$  is  $E^k$ , that is,  $(R^k)^* = E^k$ .

**Pyramid Transform on the Curved Manifold.** From a region  $\mathbf{D} \subset \mathbb{R}^n$  to a region  $\Gamma_D$  on a curved manifold  $\Gamma$ , we define a one-to-one mapping  $\xi = \phi(\mathbf{x})$  as the parameterisation on  $\Gamma_D$ . Using parameterisation on  $\Gamma$ , we define the downsampling by the factor  $\sigma$  on  $\Gamma_D$  as

$$\sigma[\xi] = \phi(\sigma\mathbf{x}), \sigma\mathbf{x} \in \mathbf{D}. \quad (24)$$

Figure 2 shows a process of downsampling on a two-dimensional manifold  $\Gamma_D$  in three-dimensional Euclidean space  $\mathbb{R}^3$ . The bijection  $\xi = \phi(\mathbf{x})$  transforms a fine

grid in (a) and a coarse grid in (b) on  $\mathbb{R}^2$  to a fine grid (d) and a coarse grid in (e) on  $\mathbb{M}^2$ , respectively. From the coarse grid (e) on  $\mathbb{M}^2$ , we can generate a shrunken grid (f), using the transform from (b) to (c).

**Definition 2.** *The pyramid transform on a manifold is generally described as*

$$g(\boldsymbol{\xi}) = f(\sigma[\boldsymbol{\xi}], \tau), \quad \frac{\partial f}{\partial \tau} = \Delta_{\Gamma} f(\boldsymbol{\xi}, \tau), \quad (25)$$

where  $\Delta_{\Gamma}$  is the Laplace-Beltrami operator on the manifold.

On the unit sphere  $\mathbb{S}^2$  centred at the origin in three-dimensional Euclidean space  $\mathbb{R}^3$ , the vector

$$\omega(\phi, \theta) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta) \quad (26)$$

with  $\phi \in [0, 2\pi)$ ,  $\theta \in [0, \pi]$  satisfies the relation  $\omega(\phi + \pi, \pi - \theta) = \omega(\phi, \theta)$ .

The scale image  $f(\phi, \theta, \tau)$  of the image  $f(\phi, \theta) : \mathbb{S}^2 \rightarrow \mathbb{R}$  is defined as the solution of the linear heat equation

$$\frac{\partial}{\partial \tau} f(\phi, \theta, \tau) = \Delta_{\mathbb{S}^2} f(\phi, \theta, \tau), \quad f(\phi, \theta, 0) = f(\phi, \theta), \quad (27)$$

where

$$\Delta_{\mathbb{S}^2} := \frac{\partial^2}{\partial \theta^2} + \frac{1}{\tan \theta} \frac{\partial}{\partial \theta} + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \phi^2}. \quad (28)$$

On  $\mathbb{S}^2$ ,  $f(\phi, \theta)$  is expressed by the spherical harmonic series

$$f(\phi, \theta) = \sum_{l=0}^{\infty} \sum_{m=0}^l c_l^m Y_l^m(\phi, \theta), \quad (29)$$

where

$$c_l^m = \int_{\mathbb{S}^2} f(\phi, \theta) \overline{Y_l^m(\phi, \theta)} \sin \theta d\phi d\theta. \quad (30)$$

The Gaussian scale image  $f(\phi, \theta, \tau)$  of the scale  $\tau$  is expressed as

$$f(\phi, \theta, \tau) = \sum_{l=0}^{\infty} \sum_{m=0}^l \left( c_l^m e^{-l(l+1)\tau} \right) Y_l^m(\phi, \theta). \quad (31)$$

As a generalisation of the Gaussian pyramid transform, we define the pyramid transform on the sphere as follows.

**Definition 3.** <sup>3</sup>*The Gaussian pyramid transform on the sphere with the factor  $\sigma$  is*

$$R_{\sigma} f(\phi, \theta) = f(\sigma\phi, \sigma\theta, \tau), \quad (32)$$

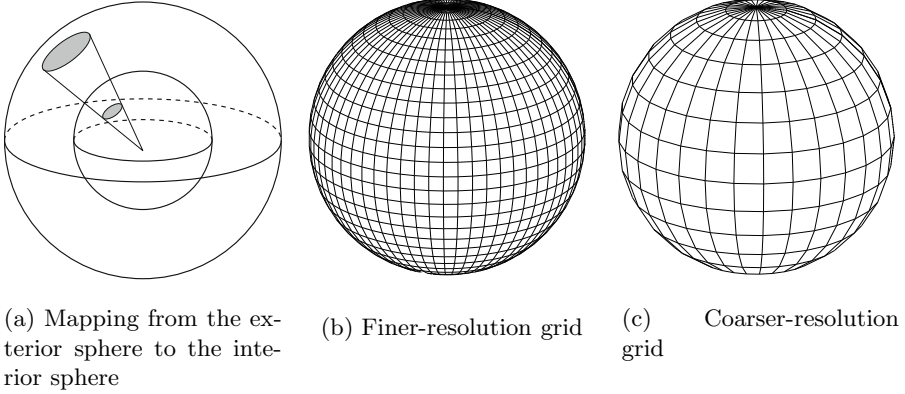
where  $0 \leq \sigma\theta \leq \pi$  and  $0 \leq \sigma\phi \leq 2\pi$ , for an appropriate positive constant  $\tau$ .

<sup>3</sup> Setting  $\sigma\mathbf{x} = (\sin \sigma\theta \cos \sigma\phi, \sin \sigma\theta \sin \sigma\phi, \cos \sigma\theta)$ , the operation is expressed as

$$R_{\sigma} f(\mathbf{x}) = \frac{1}{2\pi} \int_{\text{SO}(3)} f(\mathbf{R}\sigma[\mathbf{x}]) G(\mathbf{R}^{-1}\mathbf{y}, \tau) d\mathbf{R}, \quad |\mathbf{y}| = 1,$$

for the spherical Gaussian kernel  $G(\mathbf{x}, \tau)$ .

Figure 3 illustrates the pyramid transform on the sphere. Figure 3(a) shows that the mapping from the exterior sphere to the interior sphere defines the spherical pyramid transformation. Figures 3(b) and 3(c) show the fine- and coarse-resolution grids on the sphere, respectively. The image on the coarse-resolution grid is generated from the image on the fine-resolution grid by smoothing and downsampling.



**Fig. 3.** Pyramid transformation. (a) The mapping from the exterior sphere to the interior sphere defines the spherical pyramid transformation. (b) Finer-resolution grid on the sphere. (c) Coarser-resolution grid on the sphere. The interior and exterior spheres are expressed using the same radii.

On  $\mathbb{S}^2$ , the sampling  $I(i, j)$  of  $f(\phi, \theta)$  is defined as

$$I(i, j) = f(i\Delta_\phi, j\Delta_\theta), \quad 0 \leq i \leq 2N - 1, \quad 0 \leq j \leq N - 1, \quad (33)$$

where  $\Delta_\phi = \Delta_\theta = \pi/N$  for a positive integer  $N$ . The downsampling operation of factor 2 on the unit sphere is

$$I(i, j) = f(i(2\Delta_\phi), j(2\Delta_\theta)), \quad 0 \leq i \leq N - 1, \quad 0 \leq j \leq \left\lfloor \frac{N}{2} \right\rfloor - 1, \quad (34)$$

where  $\Delta_\phi = \Delta_\theta = \pi/N$ . The image pyramid is the sequence of images  $I^0, I^2, \dots, I^n$ , where  $I^0 = I$ .  $I^i$  is the reduced image of  $I^{i-1}$ .

### 3 Multiresolution Optical Flow Computation

**Optical Flow Computation.** For a spatiotemporal image  $f(\mathbf{x}, t)$ ,  $\mathbf{x} = (x, y)^\top$ , the optical flow vector  $\mathbf{u} = \dot{\mathbf{x}} = (\dot{x}, \dot{y})^\top$ , for  $\dot{x} = u = u(x, y)$  and  $\dot{y} = v = v(x, y)$ , of each point  $\mathbf{x} = (x, y)^\top$  is the solution of the singular equation

$$f_x u + f_y v + f_t = \nabla f^\top \mathbf{u} + \partial_t f = 0. \quad (35)$$



To solve this equation, a regularisation method [31, 43] which minimises the criterion

$$J(\mathbf{u}) = \int_{\mathbf{R}^2} \{(\nabla f^\top \mathbf{u} + \partial_t f)^2 + \kappa \text{tr} \mathbf{J} \mathbf{J}^\top\} dx dy, \quad \mathbf{J} = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \quad (36)$$

is employed [4] for the regularisation parameter  $\kappa$ . The Euler-Lagrange equations of the energy functions defined by eq. (36) and the associated diffusion equation of the Euler-Lagrange equation are

$$\Delta \mathbf{u} = \frac{1}{\kappa} (\nabla f^\top \mathbf{u} + f_t) \nabla f, \quad \frac{\partial \mathbf{u}}{\partial t} = \Delta \mathbf{u} - \frac{1}{\kappa} (\nabla f^\top \mathbf{u} + f_t) \nabla f, \quad (37)$$

with the boundary condition  $\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0$  for the unit normal  $\mathbf{n}$  on the boundary. The semi-implicit discretisation of the associated diffusion equation in eq. (37) is

$$\frac{\mathbf{u}_{ij}^{(l+1)} - \mathbf{u}_{ij}^{(l)}}{\Delta \tau} = (\Delta \mathbf{u})_{ij}^{(l)} - \frac{1}{\kappa} ((\nabla f)_{ij}^\top \mathbf{u}_{ij}^{(l+1)} + \frac{1}{\kappa} (\partial_t f)_{ij} (\nabla f)_{ij}), \quad (38)$$

where  $(f)_{ij}$  is the  $ij$ th element of the sampled function  $f(\delta i, \delta j)$  of  $f(x, y)$  for the sample interval  $\delta$ .

Using the vectorisation of the array

$$\bar{\mathbf{u}} = \text{vec} \left( \text{vec} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M1} & u_{M2} & \cdots & u_{MN} \end{pmatrix}, \text{vec} \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MN} \end{pmatrix} \right), \quad (39)$$

eq. (38) is expressed as the iteration form [46]

$$\mathbf{A} \bar{\mathbf{u}}^{(l+1)} = \mathbf{P}^\top \mathbf{B} \mathbf{P} \bar{\mathbf{u}}^{(l)} + \mathbf{c}, \quad (40)$$

for

$$\mathbf{A} = \text{diag}(\mathbf{I} + \frac{\Delta \tau}{\kappa} \mathbf{S}_{mn}), \quad \mathbf{S}_{mn} = (\nabla f)_{mn} (\nabla f)_{mn}^\top, \quad (41)$$

$$\mathbf{B} = \mathbf{I} + \Delta \tau \mathbf{L}_2, \quad (42)$$

$$\mathbf{c} = \text{vec}(\mathbf{c}_{11}, \cdots, \mathbf{c}_{MN}), \quad \mathbf{c}_{ij} = \frac{\Delta \tau}{\kappa} (f_t)_{ij} (\nabla f)_{ij}. \quad (43)$$

In this iteration form,  $\mathbf{L}_2$  is the matrix expression of the discrete Laplacian [47, 48] for the vector  $\bar{\mathbf{u}}$ , that is,

$$\mathbf{L}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \mathbf{L} \quad (44)$$

for

$$\mathbf{L} = \mathbf{D} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{D}, \quad \mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix}. \quad (45)$$

and the permutation matrix  $\mathbf{P}$  satisfies the relation

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \mathbf{Q}, \quad (46)$$

where  $\mathbf{Q}$  satisfies the permutation operation such that

$$\begin{aligned} & \mathbf{Q} \left( \text{vec} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M1} & u_{M2} & \cdots & u_{MN} \end{pmatrix}, \text{vec} \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MN} \end{pmatrix} \right) \\ &= \left( \text{vec} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M1} & u_{M2} & \cdots & u_{MN} \end{pmatrix}^\top, \text{vec} \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MN} \end{pmatrix}^\top \right). \quad (47) \end{aligned}$$

For the energy functional

$$J(\mathbf{u}, n, k; f) = \int \int_{\mathbf{R}^2} \{(\nabla f^{n\top} \mathbf{u}^n + \partial_t f^n)^2 + \kappa \text{tr} \mathbf{J}^n \mathbf{J}^{n\top}\} |_{t=k} dx dy, \quad (48)$$

where

$$\mathbf{J}^n = \begin{pmatrix} u_x^n & u_y^n \\ v_x^n & v_y^n \end{pmatrix}, \quad \mathbf{u}^n = \begin{pmatrix} u^n \\ v^n \end{pmatrix}, \quad (49)$$

which is computed from the pair  $\langle R^n f(x, y, k), R^n f(x, y, k+1) \rangle$ , we define

$$\mathbf{u}_k^n = \arg \left( \min_{\mathbf{u}} J(\mathbf{u}, n, k; f) \right). \quad (50)$$

A simple method of estimating an image in a fine grid from a coarse grid is linear interpolation. Setting  $\mathbf{u}^{(l)}$  and  $f_{(l)}$  to be the optical flow vector and image, respectively, on the  $l$ -th layer in the pyramid hierarchy, the vector

$$\mathbf{u}^{(l)} = E(\mathbf{u}^{(l+1)}) \quad (51)$$

is computed by the linear interpolation on the fine grid from a sample on the coarse grid  $\mathbf{u}^{(l+1)}$ . Furthermore, the solution of the variational problem [\[12\]](#)

$$L(\mathbf{u}^{(l)}) = \int \int_{\mathbf{R}^2} \left\{ (\nabla f_{(l)}^\top \mathbf{u}^{(l)} + \partial_t f_{(l)})^2 + \gamma (\mathbf{u}^{(l)} - E(\mathbf{u}^{(l+1)}))^2 \right\} dx dy, \quad (52)$$

where  $E(\mathbf{u}) = (Eu(x, y), Ev(x, y))^\top$ , is adopted as an initial estimate of the optical flow on the  $l$ th layer [12, 44, 43]. The minimum of  $L(\mathbf{u})$  is algebraically expressed as the solution of  $2 \times 2$  matrix equations,

$$\left( \mathbf{I} + \frac{1}{\gamma} \mathbf{S}^{(l)} \right) \mathbf{u}^{(l)} = \frac{1}{\gamma} (E(\mathbf{u}^{(l+1)}) - \partial_t f_{(l)} \nabla f_{(l)}), \quad \mathbf{S}^{(l)} = \nabla f_{(l)} \nabla f_{(l)}^\top, \quad (53)$$

for each point  $(x, y)^\top$ , that is [12],

$$\mathbf{u}^{(l)} = \left( \mathbf{I} + \frac{1}{\gamma} \mathbf{S}^{(l)} \right)^{-1} \frac{1}{\gamma} (E(\mathbf{u}^{(l+1)}) - \partial_t f_{(l)} \nabla f_{(l)}), \quad (54)$$

since  $\mathbf{I} + \frac{1}{\gamma} \mathbf{S}^{(l)}$  is non-singular for  $\gamma > 0$  for each  $(m, n)^\top$ .

Using the pyramid transform, the optical-flow field is computed by the following algorithm, where  $u$  and  $w$  denote the optical-flow computational algorithm at each pyramid level and the warping operation, respectively, where  $w(f, \mathbf{u}) = f(\mathbf{x} + \mathbf{u})$ . In the algorithm,  $\mathbf{u}_t^l = (u^l, v^l)^\top$  is computed from  $\mathbf{u}_t^{l+1} = (u^{l+1}, v^{l+1})^\top$  as  $u^l = E(u^{l+1})$  and  $v^l = E(v^{l+1})$ . Furthermore,

$$w(f_{t+1}^{l-1}, \mathbf{u}_t^l) = f_{t+1}^{l-1}(\mathbf{x} - \mathbf{u}_t^l, t), \quad (55)$$

where  $f_t^l$  denotes the pyramidal representation at level  $l$  of an image  $f(x, y, t)$  at time  $t$ .

---

**Algorithm 1.** Optical Flow Computation by the Horn-Schunk Method Using Pyramids

---

**Data:**  $f_t^l, f_{t+1}^l, \quad 0 \leq l \leq$  maximum number of the layers;

**Result:**  $\mathbf{u}_t^0 = (u_t^0, v_t^0)$ ;

$l :=$  maximum number of the layers;

**while**  $l \neq 0$ , **do**

$\mathbf{u}_t^l := \arg(\min_{\mathbf{u}} J(\mathbf{u}, n, k; f))$ ;  
 $f_{t+1}^{l-1} := w(f_{t+1}^{l-1}, \mathbf{u}_t^l)$ ;  
 $l := l - 1$ ;  
**end**

---

**Spherical Optical Flow Computation.** Since the vector expression for the spatial gradient on the unit sphere is  $\nabla_{\mathbb{S}^2} = \left( \frac{\partial}{\partial \theta}, \frac{1}{\sin \theta} \frac{\partial}{\partial \phi} \right)^\top$ , for the temporal image  $f(\theta, \phi, t)$  on the unit sphere  $\mathbb{S}^2$ , the total derivative is

$$\frac{d}{dt} f = \frac{\partial}{\partial t} f + \frac{1}{\sin \theta} \frac{\partial}{\partial \phi} f + \frac{\partial}{\partial \theta} f. \quad (56)$$

Therefore, the solution  $\dot{\omega} = \mathbf{v} = (\dot{\theta}, \dot{\phi})^\top$  of the equation

$$\mathbf{v}^\top \nabla_{\mathbb{S}^2} f + f_t = 0 \quad (57)$$

is the optical flow of image  $f$  on the unit spherical surface  $\mathbb{S}^2$ . The computation of the optical flow from eq. (57) is an ill-posed problem. The HS criterion for

the computation of the optical flow [31] on the unit sphere is expressed as the minimisation of the functional

$$J(\dot{\theta}, \dot{\phi}) = \int_{\mathbb{S}^2} \left\{ (\mathbf{v}^\top \nabla_{\mathbb{S}^2} f + f_t)^2 + \alpha (\|\nabla_{\mathbb{S}^2} \dot{\theta}\|_2^2 + \|\nabla_{\mathbb{S}^2} \dot{\phi}\|_2^2) \right\} \sin \theta d\theta d\phi, \quad (58)$$

where the  $L_2$  norm on the unit sphere is

$$\|f(\theta, \phi)\|_2^2 = \frac{1}{4\pi^2} \int_{\mathbb{S}^2} |f(\theta, \phi)|^2 \sin \theta d\theta d\phi.$$

The system of the Euler-Lagrange equations in eq. (58) is

$$\begin{aligned} \nabla_{\mathbb{S}^2}^\top \cdot \nabla_{\mathbb{S}^2} \dot{\theta} &= \frac{1}{\alpha} \frac{\partial f}{\partial \theta} \left( \frac{\partial f}{\partial \theta} \dot{\theta} + \frac{1}{\sin \theta} \frac{\partial f}{\partial \phi} \dot{\phi} + \frac{\partial f}{\partial t} \right), \\ \nabla_{\mathbb{S}^2}^\top \cdot \nabla_{\mathbb{S}^2} \dot{\phi} &= \frac{1}{\alpha \sin \theta} \frac{\partial f}{\partial \phi} \left( \frac{\partial f}{\partial \theta} \dot{\theta} + \frac{1}{\sin \theta} \frac{\partial f}{\partial \phi} \dot{\phi} + \frac{\partial f}{\partial t} \right). \end{aligned} \quad (59)$$

The associated diffusion equations of the Euler-Lagrange equations for the minimiser of eq. (58) are

$$\begin{aligned} \frac{\partial \dot{\theta}}{\partial \tau} &= \nabla_{\mathbb{S}^2}^\top \cdot \nabla_{\mathbb{S}^2} \dot{\theta} - \frac{1}{\alpha} \frac{\partial f}{\partial \theta} \left( \frac{\partial f}{\partial \theta} \dot{\theta} + \frac{1}{\sin \theta} \frac{\partial f}{\partial \phi} \dot{\phi} + \frac{\partial f}{\partial t} \right), \\ \frac{\partial \dot{\phi}}{\partial \tau} &= \nabla_{\mathbb{S}^2}^\top \cdot \nabla_{\mathbb{S}^2} \dot{\phi} - \frac{1}{\alpha \sin \theta} \frac{\partial f}{\partial \phi} \left( \frac{\partial f}{\partial \theta} \dot{\theta} + \frac{1}{\sin \theta} \frac{\partial f}{\partial \phi} \dot{\phi} + \frac{\partial f}{\partial t} \right). \end{aligned} \quad (60)$$

Therefore, from

$$\begin{aligned} \frac{\dot{\theta}^{n+1} - \dot{\theta}^n}{\Delta \tau} &= \nabla_{\mathbb{S}^2}^\top \nabla_{\mathbb{S}^2} \dot{\theta}^n - \frac{1}{\alpha} \frac{\partial f}{\partial \theta} (\nabla_{\mathbb{S}^2} f^\top \mathbf{v} + f_t), \\ \frac{\dot{\phi}^{n+1} - \dot{\phi}^n}{\Delta \tau} &= \nabla_{\mathbb{S}^2}^\top \nabla_{\mathbb{S}^2} \dot{\phi}^n - \frac{1}{\alpha \sin \theta} \frac{\partial f}{\partial \phi} (\nabla_{\mathbb{S}^2} f^\top \mathbf{v} + f_t), \end{aligned}$$

setting  $\mathbf{v} = (\dot{\theta}, \dot{\phi})^\top$ , we have the iteration form

$$(\mathbf{I} + \frac{\Delta \tau}{\alpha} \mathbf{S}_{\mathbb{S}^2}) \mathbf{v}^{(n+1)} = (\mathbf{I} + \Delta \tau \nabla_{\mathbb{S}^2}^\top \cdot \nabla_{\mathbb{S}^2}) \mathbf{v}^{(n)} + \frac{1}{\alpha} f_t \nabla_{\mathbb{S}^2} f, \quad (61)$$

where  $\mathbf{S}_{\mathbb{S}^2} = \nabla_{\mathbb{S}^2} f \nabla_{\mathbb{S}^2} f^\top$  is the structure tensor of the spherical function with the condition  $\nabla_{\mathbb{S}^2} \dot{\theta}|_{\theta=0, \pi}$ .

On  $\mathbb{S}^2$ , the sampling  $I(i, j)$  of  $f(\phi, \theta)$  is defined as

$$I(i, j) = f(i\Delta_\phi, j\Delta_\theta), \quad 0 \leq i \leq 2N - 1, \quad 0 \leq j \leq N - 1, \quad (62)$$

where  $\Delta_\phi = \Delta_\theta = \pi/N$  for a positive integer  $N$ . Therefore, the downsampling operation of factor  $2^k$  for a non-negative integer  $k$  on the unit sphere is

$$I^k(i, j) = f(i(2^k \Delta_\phi), j(2^k \Delta_\theta)) \quad (63)$$

**Algorithm 2.** PYRAMID\_OPTICALFLOW( $I, J, n$ )

---

```

Input:  $I$ : frame image
Input:  $J$ : next frame of  $I$ 
Input:  $n$ : number of levels of pyramid
Result: optical flow field between images  $I := f(\phi, \theta, t)$  and  $J := f(\phi, \theta, t + 1)$ 
begin
  Compute image pyramid  $\{I^k\}$  and  $\{J^k\}$  for  $k = 0, \dots, (n - 1)$  from  $I$  and  $J$ ,
  respectively;
   $\mathbf{v}^n \leftarrow \mathbf{0}$ ;
   $k \leftarrow n - 1$ ;
  repeat
     $\mathbf{v}^i \leftarrow \text{OPTICALFLOW}(I^k, J^k, \text{EXPAND}(\mathbf{v}^{i+1}))$ ;
     $k \leftarrow k - 1$ ;
  until  $k \geq 0$ ;
  return  $\mathbf{v}^0$ 
end

```

---

for  $0 \leq i \leq \lfloor \frac{N}{2^k} \rfloor - 1$  and  $0 \leq j \leq \lfloor \frac{N}{2^k} \rfloor - 1$ , where  $\Delta_\phi = \Delta_\theta = \pi/N$ .

The pyramid images on the unit sphere are the sequence of images  $I^0, I^1, \dots, I^n$ , where  $I^0 = I$  and

$$I^k(i, j) = R^k f(i\Delta_\phi, j\Delta_\theta, \mu(k) \times \tau_0), \quad k \geq 2 \quad (64)$$

for a positive constant  $\tau_0$  and an appropriate positive increasing function<sup>5</sup>  $g(k)$ .

For a pair of image frames  $I := f(\phi, \theta, t)$  and  $J := f(\phi, \theta, t + 1)$ , setting  $f_t := I - J$ , Algorithm 2 is the optical flow computation on the unit sphere with the pyramid transform.

## 4 Mathematical Properties of Algorithm

**Lipschitz Motion.** Using Lipschitz continuity, we define the continuity of the optical flow field such that

$$|\mathbf{u}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}, t - T)| \leq C_0, \quad (65)$$

$$|\mathbf{u}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}, t - T)| \leq C_1 T, \quad (66)$$

$$|\mathbf{u}(\mathbf{x}, t) - \mathbf{u}(\mathbf{y}, t)| \leq C_2 |\mathbf{x} - \mathbf{y}|, \quad (67)$$

$$|\mathbf{u}(\mathbf{x}, t) - \mathbf{u}(\mathbf{y}, t - T)| \leq (C_1 T + C_2 |\mathbf{x} - \mathbf{y}|), \quad (68)$$

for positive constants  $C_0, C_1$  and  $C_2$ . Equations (65) and (66) imply that the optical flow vector satisfies the total smooth and the Lipschitz continuity conditions in the temporal domain, respectively. Furthermore, eq. (67) implies that the optical flow vector satisfies the Lipschitz continuity condition in the space

<sup>5</sup> The function  $\mu(k)$  satisfies the condition  $\mu(k_1) \leq \mu(k_2)$  if  $1 < k_1 \leq k_2$ . An example is  $\mu(k) = 10^k$ .

domain. Moreover, eq. (67) implies that the optical flow vector satisfies the Lipschitz continuity condition in the spatio-temporal domain. We call the motion field which satisfies these conditions the Lipschitz motion.

Setting the average motion field between time  $t$  and  $(t + T)$  to be

$$\bar{\mathbf{u}}_T = \frac{1}{T} \int_t^{t+T} \mathbf{u}(x, y, s) ds, \quad (69)$$

eq. (65) derives the relations

$$|\bar{\mathbf{u}}_T| \leq c_{00}, \quad (70)$$

$$|\bar{\mathbf{u}}_T - \mathbf{u}| \leq c_{01}, \quad (71)$$

for positive constants  $c_{00}$  and  $c_{01}$ . Furthermore, equations (66), (67), and (68) derive the relations

$$|\partial_t \mathbf{u}| \leq c_1, \quad (72)$$

$$|\nabla \mathbf{u}| \leq c_2, \quad (73)$$

$$|\partial_t \nabla \mathbf{u}| \leq c_3, \quad (74)$$

for positive constants  $c_1$ ,  $c_1$  and  $c_2$ . Moreover, we have the relation

$$\frac{1}{|\Omega|} \int_{\Omega} |\nabla \mathbf{u}| dx dy \leq c_4 \quad (75)$$

for a positive constant  $c_4$ , where  $\Omega$  is a finite region around each point and  $|\Omega|$  is the area measure of the region.

Setting  $M(\cdot, \cdot)$  to be an appropriate measure between two fields such as angles or norms between two fields, these continuity conditions imply that

$$M(\mathbf{u}_{a,a+1}, \mathbf{u}_{a+1,a+2}) < \epsilon_1, \quad M(\mathbf{u}_{a,a+1}, \mathbf{u}_{a,a+k}) < \epsilon_2, \quad (76)$$

where  $\mathbf{u}_{ab}$  ( $a < b$ ) is the optical flow field computed between images  $f(\cdot, a)$  and  $f(\cdot, b)$ .

**Minimisation as a Series of Convex Problems.** Multiresolution optical-flow computation establishes an algorithm which guarantees the relation

$$\lim_{n \rightarrow 0} \mathbf{u}^n = \mathbf{u} \quad (77)$$

for each time. Since  $J(\mathbf{u}, n, t; f)$  is a convex functional for a fixed  $f$ , this functional satisfies the relation

$$J(\mathbf{u}^n, n, t; f) < J(\mathbf{u}, n, t; f) \quad (78)$$

for  $\mathbf{u}^n \neq \mathbf{u}$ . Therefore, we have the relation

$$J(\mathbf{u}^{(n-1)}, n-1, t; f) \leq J(\mathbf{u}^n, n-1, t; f). \quad (79)$$

This relation implies that it is possible to generate a sequence which reaches  $E(\mathbf{u}_0, 0, t; f)$  from  $E(\mathbf{u}_n, n - 1, t; f)$ , since

$$\begin{aligned}
 J(\mathbf{u}^{(n-1)}, n - 1, t; f) &\leq J(E(\mathbf{u}^n), n - 1, t; f) \\
 J(\mathbf{u}^{(n-2)}, n - 2, t; f) &\leq J(E(\mathbf{u}^{(n-1)}), n - 2, t; f) \\
 &\vdots \\
 J(\mathbf{u}^1, 1, t; f) &\leq J(E(\mathbf{u}^2), 1, t; f) \\
 J(\mathbf{u}^0, 0, t; f) &\leq J(E(\mathbf{u}^1), 0, t; f)
 \end{aligned} \tag{80}$$

for a fixed  $f$ , setting  $\mathbf{u}^{(n-1)} = E(\mathbf{u}^n) + \mathbf{d}^{(n-1)}$ , and for a fixed time  $t$ , where  $|\mathbf{d}^{(n-1)}| \ll |\mathbf{u}^{(n-1)}|$ . Figure 4 illustrates a geometrical interpretation of eq. (80). The minimum of a fixed resolution derives an approximation of the minimum for the next finer resolution for optical flow computation.

**Convergence Conditions.** Using the conditions defined by eqs. (70) and (72) we introduce the following definitions for the optical-flow vector  $\mathbf{u}$ .

**Definition 4.** *If a flow vector satisfy the condition  $|\mathbf{u}|_m \leq \alpha$ , for  $|\mathbf{u}|_m = \max_{\mathbf{x} \in \mathbb{R}^2} |u(\mathbf{x})|$  we call the flow vector is  $\alpha$ -stationary.*

**Definition 5.** *For a sufficiently small positive constant  $\beta$ , if the optical-flow vector  $\mathbf{u}$  satisfies the relation  $|\frac{\partial \mathbf{u}}{\partial t}|_m \leq \beta$ , we call the flow field  $\beta$ -time stationary. In particular, if  $\beta = 0$ , the flow field is stationary in the temporal domain.*

For the area measure  $|\Omega|$  of a tessellated region, a possible selection for  $\alpha$  is on the order of  $\sqrt{|\Omega|}$ , that is, we can set  $\alpha = c\sqrt{|\Omega|}$  for  $0 < c \leq 1$ . In reference [45],  $\beta$  is also assumed to be  $\beta = c\sqrt[3]{|\Omega| \times T}$ ,  $0 < c \leq 1$ , where  $T$  is the time frame interval and is usually set as 1. These conditions give a mathematical description of the temporal smoothness of the optical flow vector at each point.

Next we introduce the spatial smoothness assumption applied to the optical flow vectors using the relations defined by eqs. (71), (73) and (75).

**Definition 6.** *For a sufficiently small positive constant  $\gamma$ , if the optical-flow vector  $\mathbf{u}$  satisfies the relation  $|\nabla \mathbf{u}| = \sqrt{\text{tr} \mathbf{J} \mathbf{J}^T} \leq \gamma$  in domain  $\Omega$ , we call flow field  $\gamma$ -spatial stationary.*

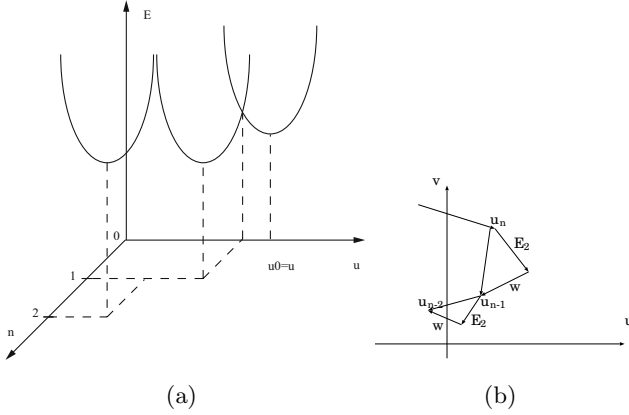
Furthermore, we introduce the cross-layer relation of a flow vector.

**Definition 7.** *For  $\overline{\mathbf{u}^n} = \frac{1}{|\Omega|} \int_{\Omega} \mathbf{u}^n d\mathbf{x}$ , if*

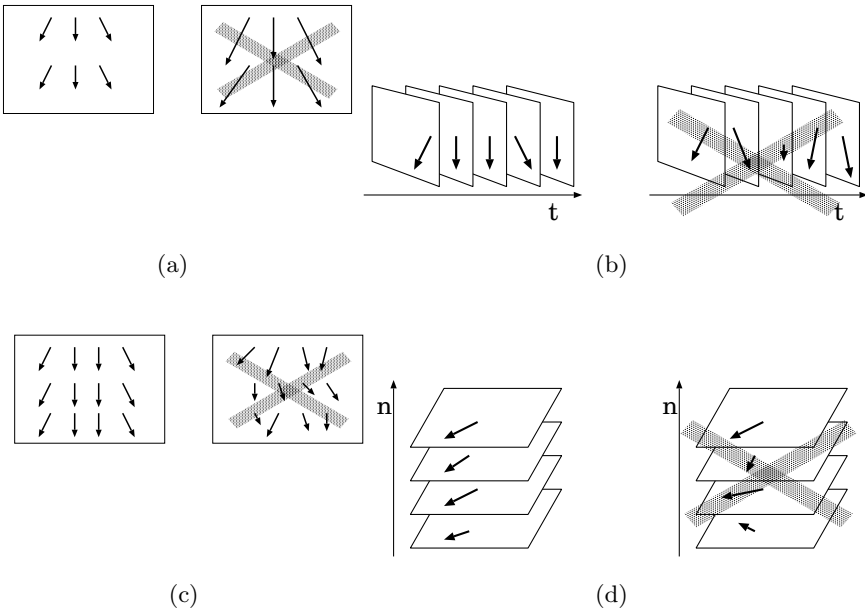
$$|\mathbf{u}^n - \overline{\mathbf{u}^n}|_m \leq \frac{\delta}{3}, \quad |\overline{\mathbf{u}^{(n-1)}} - E(\overline{\mathbf{u}^n})|_m \leq \frac{\delta}{3}, \quad |\mathbf{u}^{(n-1)} - \overline{\mathbf{u}^{(n-1)}}|_m \leq \frac{\delta}{3}, \tag{81}$$

*we call  $\mathbf{u}^n$   $\delta$ -layer stationary.*

From Definition 4, we have the relation  $|\mathbf{u}^n - \overline{\mathbf{u}^n}|_m \leq 2\alpha$ . Therefore, the second condition controls the interlayer continuity of the flow vectors. It is possible to assume  $\delta = c\sqrt{|\Omega|}$ ,  $0 < c \leq 1$ .



**Fig. 4.** Convergence geometry. (a) The minimum of a fixed resolution derives an approximation of the minimum for the next finer resolution for optical flow computation. (b) In classical multiresolution optical flow computation, the approximate solution converges in each frame.



**Fig. 5.** Convergence conditions for optical flow on different layers. (a) The displacement of the  $\alpha$ -stationary optical flow is small. (b) The  $\beta$ -stationary optical flow is smooth in the time domain. (c) The  $\gamma$ -stationary optical flow is smooth. (d) The  $\delta$ -stationary optical flow is smooth across the layers.



Figure 5 shows geometrical interpretations of these definitions. (a) The displacement of the  $\alpha$ -stationary optical flow is small. (b) The  $\beta$ -stationary optical flow is smooth in the time domain. (c) The  $\gamma$ -stationary optical flow is smooth. (d) The  $\delta$ -stationary optical flow is smooth across the layers.

Setting  $\Delta_n \mathbf{u}^n(\mathbf{x}, t) = E(\mathbf{u}^n(\mathbf{x}, t)) - \mathbf{u}^{(n-1)}(\mathbf{x}, t)$ , we have the following lemma.

**Lemma 1.** *If  $\mathbf{u}^n$  is  $\delta$ -layer stationary, the relation  $|\Delta_n \mathbf{u}^n(\mathbf{x}, t)|_m \leq \delta$  is satisfied.*

If  $\mathbf{u}^{(n-1)}$  is  $\alpha$ -stationary, that is,  $|\mathbf{u}^{(n-1)}| \leq \alpha$ , we have the relation  $|\overline{\mathbf{u}^n}|_m \leq \frac{\alpha}{\mathcal{D}}$ . Since

$$|E(\mathbf{u}^{(n-1)}) - \mathbf{u}^n|_m \leq |\mathbf{u}^{(n-1)} - \overline{\mathbf{u}^{(n-1)}}|_m + |E(\overline{\mathbf{u}^{(n-1)}}) - \overline{\mathbf{u}^n}|_m + |\mathbf{u}^n - \overline{\mathbf{u}^n}|_m,$$

we have the relation  $|\mathbf{u}^{(n-1)} - E_2(\mathbf{u}^n)|_m \leq \delta$ .

This relation leads to the next theorem.

**Theorem 1.** *If the optical flow vectors are  $\delta$ -layer stationary, the conventional pyramid algorithm converges.*

Furthermore, since

$$\begin{aligned} \mathbf{u}_{t+1}^{(n-1)} - E(\mathbf{u}_t^n) &= \mathbf{u}_{(t+1)}^{(n-1)} - E(\mathbf{u}_{(t+1)}^n) + E(\mathbf{u}_{(t+1)}^n) - E(\mathbf{u}_t^n) \\ &= -\Delta_n \mathbf{u}_{(t+1)}^n + E\left(\frac{\partial}{\partial t} \mathbf{u}_t^n\right), \end{aligned} \quad (82)$$

we have the relation

$$|\mathbf{u}_{(t+1)}^{(n-1)} - E(\mathbf{u}_t^n)|_m \leq \delta + \left| E\left(\frac{\partial}{\partial t} \mathbf{u}_t^n\right) \right|_m \leq \beta + \delta. \quad (83)$$

This relation leads to the next theorem.

**Theorem 2.** *Setting  $R$  to be an image transform used to derive a low-resolution image from an image  $f$ , the low-resolution image is expressed as  $Rf$ . Then,  $\delta$  is sufficiently small, and the motion is  $\beta$ -time stationary for a sufficiently small constant, and Algorithm 1 proposed in the previous section converges.*

This theorem guarantees the convergence of the multiresolution optical flow for a Gaussian pyramid transform.

## 5 Numerical Results

### 5.1 Planar Images

We evaluate our method using the four test image sequences shown in Fig. 6. In experiments, optical flows between the first and  $i$ th frames in the sequences are

computed both with and without a scale-space pyramid transform. The pyramid of an image is constructed from the zeroth level, which is the same as the original image, to the fourth level by Gaussian blurring with variance  $\sigma = \sqrt{2}\Delta$ , where  $\Delta$  is the pixel size and is set to 1, followed by downsampling. Therefore, the area of the image at the  $i$ th level is one-quarter of that at the  $(i - 1)$ th level.

Let  $u_{\text{pyr}}^k(x, y, t)$  and  $u_{\text{ord}}^k(x, y, t)$  be the optical flows between  $f(x, y, t)$  and  $f(x, y, t + k)$ , with and without a pyramid transform, respectively. We compare the average optical flow in a long interval,  $\frac{1}{k}u^k(x, y, t)$  for  $k = 2, \dots, n$ , with the optical flow in a short interval,  $u^1(x, y, t)$ , where  $u^k = u_{\text{pyr}}^k, u_{\text{ord}}^k$  and  $n + 1$  is the number of images in the sequence shown in Table 1. The statistical results for each sequence are shown in Figs. 7-10.

In general, the HS-type optical flow computation cannot to be applied for a long-interval sequence since the displacement of motion is too long and violates the condition of the optical flow constraint.

By using the pyramid-based computation, the errors for  $k = 2, \dots, 5$  are improved in all sequences.

For the sequences Old Marbled Block and Yosemite, the optical flows are improved by using the pyramid method, while there is little difference in the Daimler sequence, because the motion in the sequence is constant.

The Metronome sequence contains a quick-moving object, and the computation of optical flow fails in both methods, which is indicated by the large variance of the angle error.

**Table 1.** Image sequences used in the experiments

Sequence	# Images	Size	Reference
Daimler (EISATS Set2 Left)	10	$512 \times 512$	
Metronome	10	$896 \times 1072$	
Old Marbled Block	10	$512 \times 512$	
Yosemite	8	$316 \times 252$	

## 5.2 Spherical Images

Figure 11 shows the first frames and the corresponding scale images in the panoramic projection of real and synthetic spherical image sequences captured by a moving omnidirectional camera passing through a corridor. The size of the original image is  $256 \times 128$  pixels in the equi-rectangular projection map. Therefore, the sizes of the images in the first and second levels are  $128 \times 64$  and  $64 \times 32$  pixels, respectively. The scale parameters for the first and second transforms are  $\tau_1 = 0.0001$  and  $\tau_2 = 10 \times \tau_1$ , respectively. In the experiments, we set the maximum order of the spherical harmonic series to  $l_{\text{max}} = 127$ . The translation is 1cm per frame in the  $\phi = 180^\circ$  direction, which is the horizontal center of the image.

Daimler



Metronome



Old Marbled Block



Yosemite



**Fig. 6.** First images in the sequences used in the experiments and their pyramid transforms. The leftmost image is the 0th level, which is the original image, and the rightmost image is the 4th level.

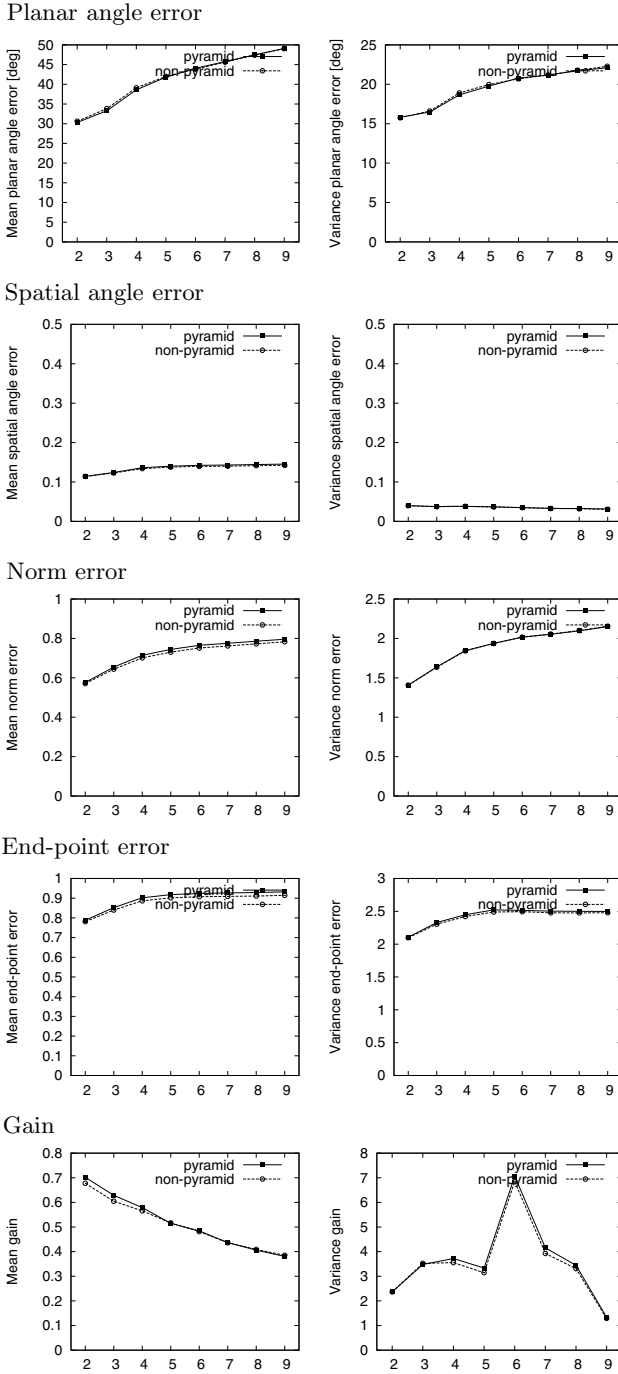
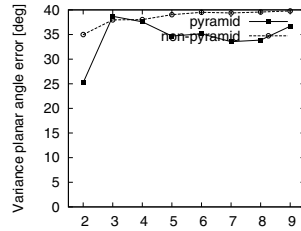
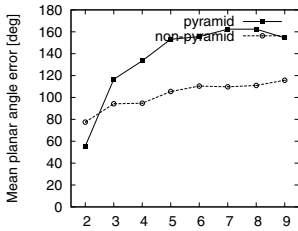
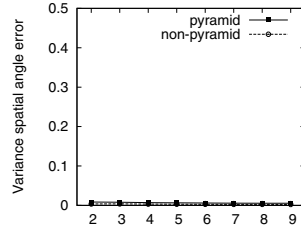
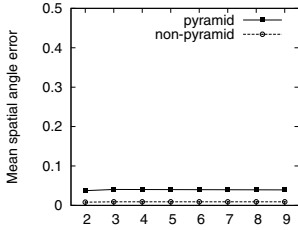


Fig. 7. Statistics for each interval of optical flow (Daimler)

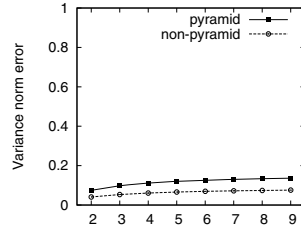
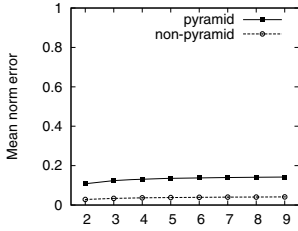
## Planar angle error



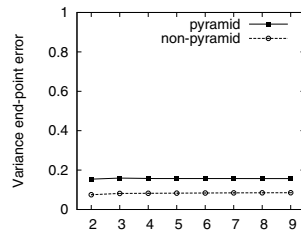
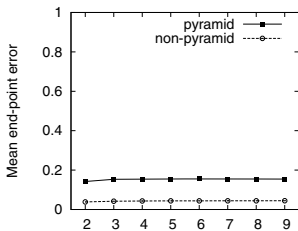
## Spatial angle error



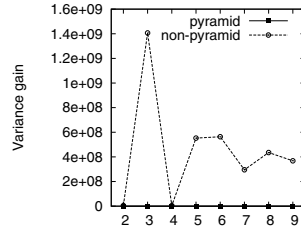
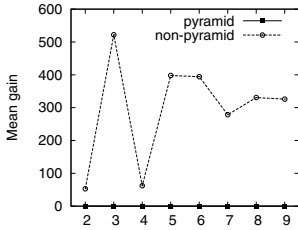
## Norm error



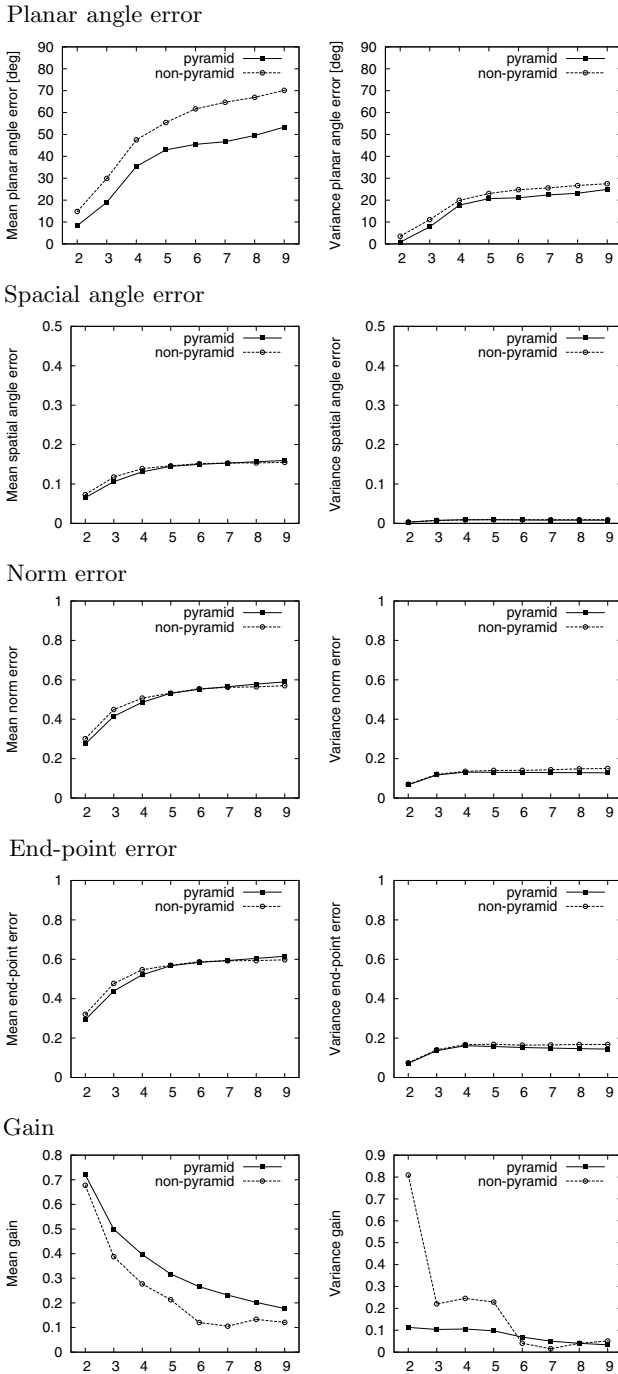
## End-point error



## Gain

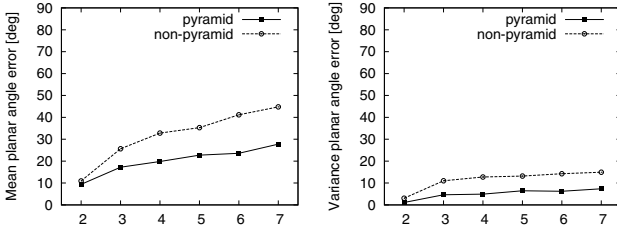


**Fig. 8.** Statistics for each interval of optical flow (Metronome)

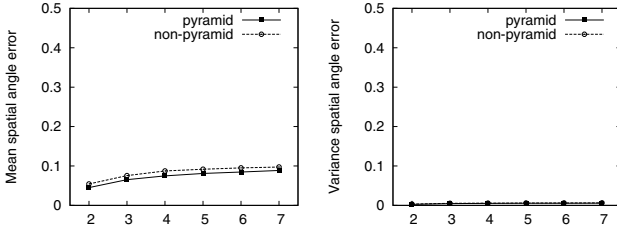


**Fig. 9.** Statistics for each interval of optical flow (Old Marbled Block)

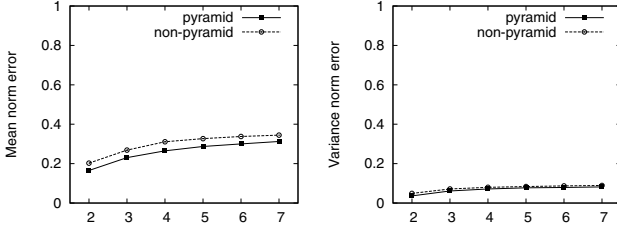
Planar angle error



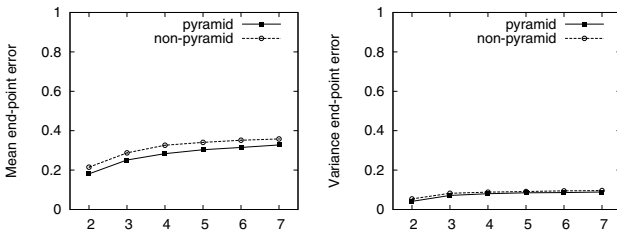
Spatial angle error



Norm error



End-point error



Gain

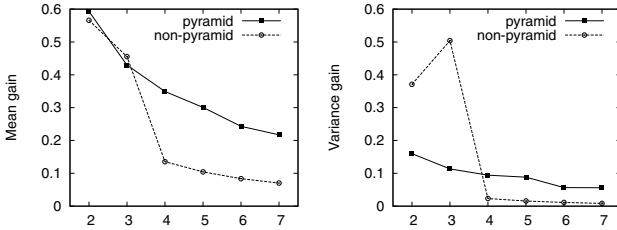


Fig. 10. Statistics for each interval of optical flow (Yosemite)

Figures 12 and 13 show the computed optical flow with and without the pyramid method for two frame intervals for the real- and synthetic-image sequences, respectively.

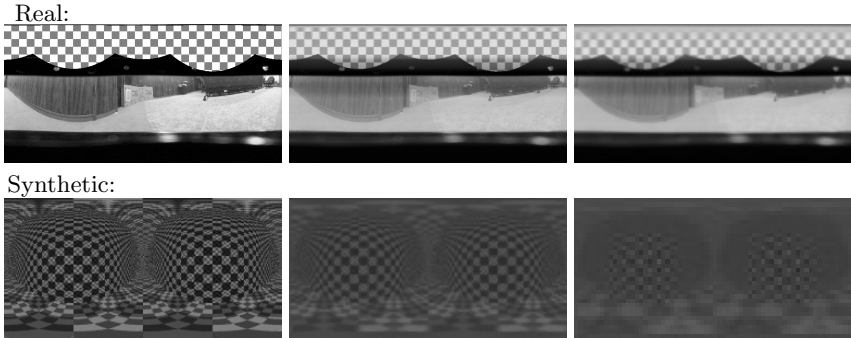


Fig. 11. Real- and synthetic-image sequences

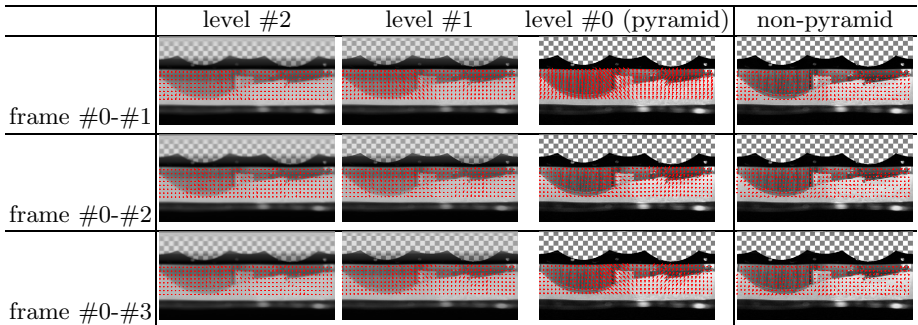


Fig. 12. Results for the real-image sequence. The HS parameter is set to  $\alpha = 0.5$ . The X-axis is the interval between two frames.

Since the optical flow vector defines an infinitesimal displacement, the optical flow  $\mathbf{v} = (\dot{\theta}, \dot{\phi})^\top$  of the spherical image defines the infinitesimal rotation

$$\mathbf{R} = \begin{pmatrix} 0 & -\dot{\phi} & \dot{\theta} \\ \dot{\phi} & 0 & 0 \\ -\dot{\theta} & 0 & 0 \end{pmatrix}, \quad (84)$$

which is equivalent to the vector  $\mathbf{u} = (0, \dot{\theta}, \dot{\phi})^\top \in \mathbf{R}^3$ .

To evaluate the optical flows, we use the norm and angle errors. For all points  $(\phi, \theta)$ , the two flow vectors on the tangent plane are compared in terms of norm and angle errors. The absolute norm error between  $\mathbf{u}$  and  $\mathbf{v}$  is measured as

$$n = \|\|\mathbf{u}\| - \|\mathbf{v}\|\|, \quad (85)$$



	level #2	level #1	level #0 (pyramid)	non-pyramid
frame #0-#1				
frame #0-#2				
frame #0-#3				

**Fig. 13.** Results for the synthetic-image sequence. The HS parameter is set to  $\alpha = 0.5$ . The X-axis is the interval between two frames.

the spatial angle error as

$$\theta = \cos^{-1} \frac{\hat{\mathbf{u}}^\top \hat{\mathbf{v}}}{\|\hat{\mathbf{u}}\| \|\hat{\mathbf{v}}\|}, \quad (86)$$

where  $\hat{\mathbf{u}} := (\mathbf{u}^\top, \Delta_\phi)^\top$  and  $\hat{\mathbf{v}} := (\mathbf{v}^\top, \Delta_\phi)^\top$ , and the planar angle error as

$$\bar{\theta} = \cos^{-1} \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}. \quad (87)$$

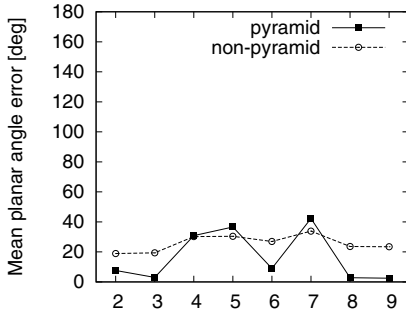
We set the planar angle error to 0 if either  $\mathbf{u}$  or  $\mathbf{v}$  is  $\mathbf{0}$ .

Figures 14 and 15 show the statistical results for difference between the short displacement flow, computed from frames  $t$  and  $t + 1$ , and the long displacement flow, computed from frames  $t$  and  $t + k$ , for the real-image and synthetic-image sequences, respectively. The figures are plotted for each  $k$ . The results show that the pyramid-based optical flow computation can compute motion with both small and large displacements.

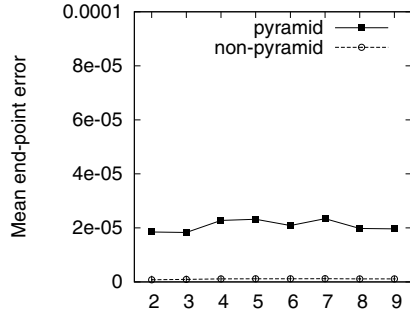
The rotation of a robot around a point causes the rotation of a spherical image around the axis perpendicular to the floor and translation on a panoramic image. The translation of the robot causes a divergent optical flow on both the spherical image and the panoramic image.

For real-image sequences, we cannot prepare ground truth for the evaluation of computed results. If the motion appearing in the captured image sequence is locally stationary, optical flow fields for a pair of successive images are stationary. Therefore, the difference between the optical flow fields for a pair of successive images is small. Using this property of the optical flow sequence, we evaluated the results using the flow field sequence computed from the real-image sequence without using ground truth.

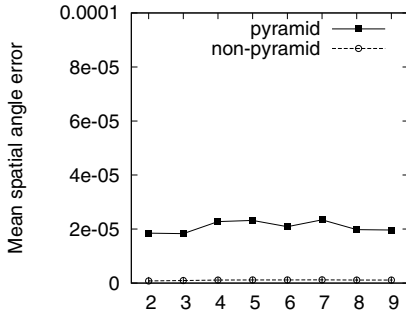
The measure  $\dot{\omega}_{i,n+i} - \dot{\omega}_{j,n+j}$  indicates the stability of the optical flow if the motion is uniform in an environment with relatively small obstacles. If  $n > 2$ , the optical flow is considered as a large displacement. The measure  $\dot{\omega}_{i,n+i} - \dot{\omega}_{j,n+j}^*$  indicates the difference between the pyramid-based and non-pyramid-based methods. The results show that the pyramid-based method computes optical flow accurately for image sequences with large displacement.



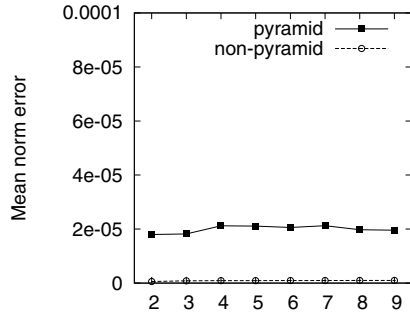
(a) Angle error (planar)



(b) End-point error

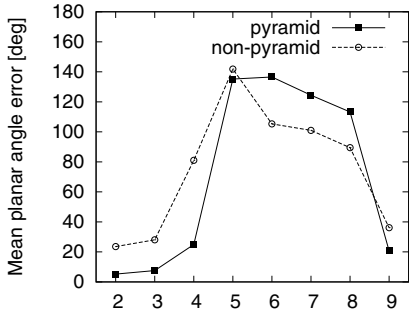


(c) Angle error (spatial)

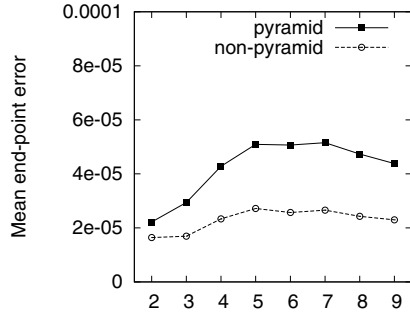


(d) Norm error

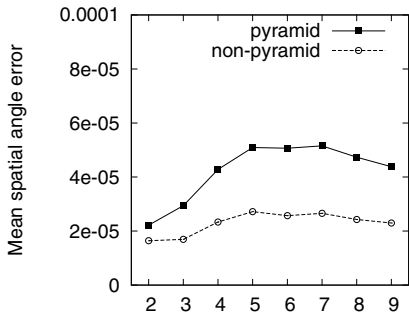
**Fig. 14.** Statistics for the real-image sequence



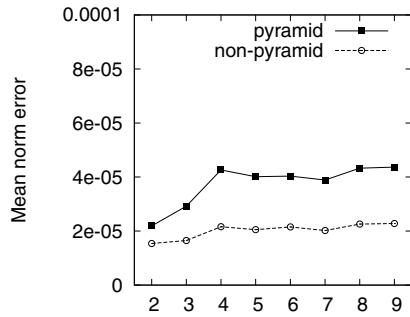
(a) Angle error (planar)



(b) End-point error



(c) Angle error (spatial)



(d) Norm error

**Fig. 15.** Statistics for the synthetic-image sequence

## 6 Conclusions

The pyramid transform is efficiently used in optical flow computation for planar images captured by pinhole camera systems, since the propagation of features from coarse sampling to fine sampling allows the computation of both large displacements in low-resolution images sampled by a coarse grid and small displacements in high-resolution images sampled by a fine grid.

The Gaussian pyramid transform on the plane is achieved by downsampling of the convolution between an image and a kernel function. Since the convolution with the Gaussian kernel is the solution of the linear diffusion equation, the Gaussian pyramid is obtained by applying downsampling to the solution of the linear diffusion equation. We have extended this idea.

In images captured by an omnidirectional imaging system, moving objects and target objects are relatively sparse, since the system images a wide-view environment in a single view. The pyramid transform compresses a wide-view image to a small image preserving the global features of the image. Therefore, pyramid transforms are suitable for the preprocessing of an omnidirectional image/image sequence. Since omnidirectional images are geometrical images on a curved manifold, we introduced the pyramid transform and a multiresolution representation on the curved manifold.

Since the real-world images captured by an imaging system mounted on a car and on a mobile robot used for navigation and understanding of the environment have no ground truth, for the evaluation of computer vision algorithms in a large real-world environment, we have introduced a method to evaluate results simultaneously for an optical flow field using the continuity assumption.

## References

1. Kimuro, Y., Nagata, T.: Image processing on an omni-directional view using a spherical hexagonal pyramid: Vanishing points extraction and hexagonal chain code. In: Proc. IROS 1995, pp. 356–361 (1995)
2. Kin, G., Sato, M.: Scale space filtering on spherical pattern. In: Proc. ICPR 1992, vol. 3, pp. 638–641 (1992)
3. Bülow, T.: Spherical diffusion for 3D surface smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1650–1654 (2004)
4. Morales, S., Klette, R.: A Third Eye for Performance Evaluation in Stereo Sequence Analysis. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 1078–1086. Springer, Heidelberg (2009)
5. Ohnishi, N., Imiya, A.: Featureless robot navigation using optical flow. *Connection Science* 17, 23–46 (2005)
6. Jolion, J.M.: Stochastic pyramid revisited. *Pattern Recognition Letters* 24, 1035–1042 (2003)
7. Jolion, J.M., Montanvert, A.: The adaptive pyramid: a framework for 2D image analysis. *CVGIP: Image Understanding* 55, 339–348 (1992)
8. Jolion, J.M., Rosenfeld, A.: An  $O(\log n)$  pyramid Hough transform. *Pattern Recognition Letters* 9, 343–349 (1989)

9. Kropatsch, W.G.: A pyramid that grows by powers of 2. *Pattern Recognition Letters* 3, 315–322 (1985)
10. Kropatsch, W.G.: Curve representations in multiple resolutions. *Pattern Recognition Letters* 6, 179–184 (1987)
11. Kropatsch, W.G.: Building irregular pyramids by dual graph contraction. In: *IEEE-Proc. Vision, Image and Signal Processing*, vol. 142(6), pp. 366–374 (1995)
12. Hwan, S., Hwang, S.-H., Lee, U.K.: A hierarchical optical flow estimation algorithm based on the interlevel motion smoothness constraint. *Pattern Recognition* 26, 939–952 (1993)
13. Witkin, A.P.: Scale space filtering. In: *Proc. of 8th IJCAI*, pp. 1019–1022 (1983)
14. Koenderink, J.J.: The structure of images. *Biological Cybernetics* 50, 363–370 (1984)
15. Zhao, N.-Y., Iijima, T.: Theory on the method of determination of view-point and field of vision during observation and measurement of figure. *IEICE Japan, Trans. D J68D*, 508–514 (1985) (in Japanese)
16. Zhao, N.-Y., Iijima, T.: A theory of feature extraction by the tree of stable view-points. *IEICE Japan, Trans. D J68D*, 1125–1135 (1985) (in Japanese)
17. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications* 31, 532–540 (1983)
18. Olkkonen, H., Pesola, P.: Gaussian pyramid wavelet transform for multiresolution analysis of images. *Graphical Models and Image Processing* 58, 394–398 (1996)
19. Weickert, J., Ishikawa, S., Imiya, A.: Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision* 10, 237–252 (1999)
20. Lindeberg, T.: *Scale-Space Theory in Computer Vision*. Kluwer, Boston (1994)
21. Duits, R., Florack, L., Graaf, J., ter Haar Romeny, B.: On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision* 20, 267–298 (2004)
22. Johansen, P., Skelboe, S., Grue, K., Andersen, J.D.: Representing signals by their toppoints in scale space. In: *Proc. International Conference on Image Analysis and Pattern Recognition*, pp. 215–217 (1986)
23. Guilherme, N.D., Avinash, C.K.: Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 237–267 (2002)
24. Franz, M.O., Mallot, H.A.: Biomimetic robot navigation. *Robotics and Autonomous Systems* 30, 133–153 (2000)
25. Franz, M.O., Chahl, J.S., Krapp, H.G.: Insect-inspired estimation of egomotion. *Neural Computation* 16, 2245–2260 (2004)
26. Green, W.E., Oh, P.Y., Barrows, G.: Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments. In: *Proc. ICRA 2004*, vol. 3, pp. 2347–2352 (2004)
27. Sobey, P.J.: Active navigation with a monocular robot. *Biological Cybernetics* 71, 433–440 (1994)
28. Vardy, A., Moller, R.: Biologically plausible visual homing methods based on optical flow techniques. *Connection Science* 17, 47–89 (2005)
29. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proc. Imaging Understanding Workshop*, pp. 121–130 (1981)
30. Chianga, M.-C., Boulton, T.E.: Efficient super-resolution via image warping. *Image and Vision Computing* 18, 761–771 (2000)
31. Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artificial Intelligence* 17, 185–203 (1981)

32. Terzopoulos, D.: Image analysis using multigrid relaxation method. *IEEE PAMI* 8, 129–139 (1986)
33. de Zeeuw, P.M.: The multigrid image transform. In: Tai, X.-C., Lie, K.A., Chan, T., Osher, S. (eds.) *Image Processing Based on Partial Differential Equations: Mathematics and Visualization*, pp. 309–324. Springer (2007)
34. Briggs, W.L., Henson, V.E., McCormick, S.F.: *A Multigrid Tutorial*, 2nd edn. SIAM (2000)
35. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 31, 333–390 (1977)
36. Trottenberg, U., Oosterlee, C.W., Schüller, A.: *Multigrid*. Academic Press (2001)
37. Larsson, J., Lien, F.S., Yee, E.: Conditional semi-coarsening multigrid algorithm for the Poisson equation on anisotropic grids. *Journal of Computational Physics* 208, 368–383 (2005)
38. Williamson, D.L.: The evolution of dynamical cores for global atmospheric models. *Journal of the Meteorological Society of Japan* 85B, 241–269 (2007)
39. Buckeridge, S., Scheichl, R.: Paralle geometric multigrid for global weather prediction. *Numerical Linear Algebra with Applications* 17, 325–342 (2010)
40. Wagner, C., Christian Wagner’s: Algebraic Multigrid Tutorial: Introduction to algebraic multigrid, Course notes of an algebraic multigrid course at the University of Heidelberg in the Wintersemester (1998/1999), <http://www.mgnet.org/mgnet-tuts.html>
41. Imiya, A., Kameda, Y., Ohnishi, N.: Decomposition and Construction of Neighbourhood Operations Using Linear Algebra. In: Coeurjolly, D., Sivignon, I., Tougne, L., Dupont, F. (eds.) *DGCI 2008*. LNCS, vol. 4992, pp. 69–80. Springer, Heidelberg (2008)
42. Strang, G., Nguyen, T.: *Wavelets and Filter Banks*. Wellesley-Cambridge Press (1996)
43. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. *ACM Computer Surveys* 26, 433–467 (1995)
44. Amiaz, T., Lubetzky, E., Kiryati, N.: Coarse to over-fine optical flow estimation. *Pattern Recognition* 40, 2496–2503 (2007)
45. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision* 2, 283–310 (1989)
46. Varga, R.S.: *Matrix Iteration Analysis*, 2nd edn. Springer (2000)
47. Demmel, J.W.: *Applied Numerical Linear Algebra*. SIAM (1997)
48. Strang, G.: *Computational Science and Engineering*. Wellesley-Cambridge Press (2007)

## Appendix

In this appendix, we assume that the multiscale grid system  $\{\Omega_i\}_{i=1}^L$  satisfies the relation

$$\Omega_0 \subset \Omega_1 \subset \Omega_2 \subset \cdots \subset \Omega_L,$$

that is,  $\{\Omega_i\}_{i=1}^{L-1}$  are the coarse grid systems and  $\Omega_L$  is the original grid system. Therefore, the reduction and expansion are transforms for discrete functions from  $\Omega_i$  to  $\Omega_{i-1}$  and from  $\Omega_{i-1}$  to  $\Omega_i$ , respectively.

We compare the pyramid-transform-based linear-equation solver (PS) and the algebraic multigrid method. Before the main comparison, we define a relaxation

**Algorithm 3.** Pyramid Transform Based Linear Equation Solver

---

```

input  $\mathbf{A}_L := \mathbf{A}$ , for  $0 \leq l \leq L - 1$ ,  $\mathbf{A}_{l-1} = \mathbf{R}\mathbf{A}_l\mathbf{R}^\top$ ,  $\mathbf{f}_{l-1} = \mathbf{R}\mathbf{f}_l$ ,
 $\mathbf{u}_0 :=$  an initial vector;
output  $\mathbf{u} := \mathbf{u}_L$ ;
for  $l := 0$  to  $L - 1$  do
   $\mathbf{v}_{l+1} := \mathbf{E}\mathbf{u}_l$ ;
   $\mathbf{d}_{l+1} := S^\mu(\mathbf{A}\mathbf{d}_{l+1} = (\mathbf{f}_{l+1} - \mathbf{A}\mathbf{v}_{l+1}))$ ;
   $\mathbf{u}_{l+1} := \mathbf{v}_{l+1} + \mathbf{d}_{l+1}$ ;
   $l := l + 1$ 

```

---

**Algorithm 4.** The V-cycle Multigrid Method

---

```

input  $\mathbf{A}_L := \mathbf{A}$ , for  $0 \leq l \leq L - 1$ ,  $\mathbf{A}_{l-1} = \mathbf{R}\mathbf{A}_l\mathbf{R}^\top$ ,  $\mathbf{f}_{l-1} = \mathbf{R}\mathbf{f}_l$ ,
 $\mathbf{u}_0 :=$  an initial vector;
output  $\mathbf{u} := \mathbf{u}_L$ ;
 $M(\mathbf{u}_l, \mathbf{f}_l, l)$ ;
begin
  if  $l = 0$  then
     $\mathbf{u}_l = \mathbf{A}_l^{-1} \mathbf{f}_l$ 
  else
     $\mathbf{u}_l := S^{\mu_1}(\mathbf{A}\mathbf{u}_l = \mathbf{f}_l)$ ;
     $\mathbf{d}_{l-1} := \mathbf{R}(\mathbf{f}_l - \mathbf{A}\mathbf{u}_l)$ ;
     $\mathbf{v}_{l-1} := 0$ ;
    call  $M(\mathbf{v}_{l-1}, \mathbf{f}_{l-1}, l - 1)$ ;
     $\mathbf{u}_l := \mathbf{u}_l + \mathbf{E}\mathbf{v}_{l-1}$ ;
     $\mathbf{u}_l := S^{\mu_2}(\mathbf{A}\mathbf{u}_l = \mathbf{f}_l)$ ;
  end

```

---

solver for the system of linear equations  $\mathbf{A}\mathbf{u} = \mathbf{f}$ , assuming that  $\mathbf{A}$  is non-singular. By selecting an appropriate invertible matrix  $\mathbf{M}$ , the iteration form

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \mathbf{M}^{-1}(\mathbf{f} - \mathbf{A}\mathbf{u}^{(n)})$$

derives the Jacobi, Gauss-Seidel or Incomplete LU decomposition methods as a linear equation solver. We define the relaxation procedure  $\mathbf{u}^\mu := S^\mu(\mathbf{A}\mathbf{u} = \mathbf{f})$  for a positive integer  $\mu$ .

Using the relaxation procedure  $\mathbf{v} := S^\mu(\mathbf{A}\mathbf{u} = \mathbf{f})$ , we have Algorithm 3. For a pre-fixed positive integer  $L$ , the V-cycle multigrid method is a recursive procedure  $M(\mathbf{u}_l, \mathbf{f}_l, l)$  in the Algorithm 4 [40]. In this procedure,  $\mathbf{u} := \mathbf{u}_L$  is the multigrid solution of the system of linear equations  $\mathbf{A}\mathbf{u} = \mathbf{f}$ .

The PS computes an estimation of the solution in a finer grid using the equation in a coarse grid and computes the correction to the estimated solution using the residual in the finer grid. On the other hand, the algebraic multigrid method computes the correction to the solution in a finer grid using the reduction of the residual in the finer grid to a coarse grid.

# Towards Feature-Based Situation Assessment for Airport Apron Video Surveillance

Ralf Dragon<sup>1</sup>, Michele Fenzi<sup>1</sup>, Wolf Siberski<sup>2</sup>,  
Bodo Rosenhahn<sup>1</sup>, and Jörn Ostermann<sup>1</sup>

<sup>1</sup> Institut für Informationsverarbeitung  
{dragon, fenzi, rosenhahn, ostermann}@tnt.uni-hannover.de  
<sup>2</sup> L3S

Leibniz Universität Hannover, Germany  
siberski@l3s.uni-hannover.de

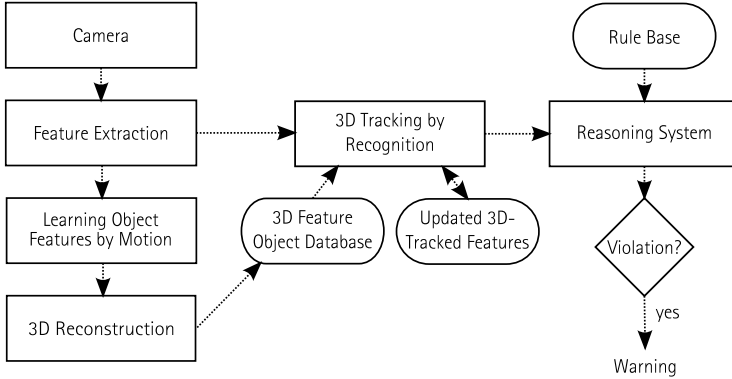
**Abstract.** We present a feature-based surveillance pipeline which, in contrast to traditional image-based methods, allows to learn a detailed description of the observed background as well as of foreground objects. The pipeline consists of motion segmentation of feature trajectories and subsequent tracking-by-recognition with updates. Furthermore, 3D object representations are learned in order to extract the 3D object pose of a later object recognition. Finally, we show how such sufficiently reliable information is inputted into a reasoning system comparing actual and nominal condition of an airport apron. By this, automatic situation assessment becomes possible in a manageable and reliable way.

## 1 Introduction

Video surveillance is a field in which manual interpretation of camera images dominates. Although it is known that the human assessment of video material is a fatiguing task with a short attention span of approximately 20 minutes [19], computer assistance for operators is still at a very basic level: The usual assistance is activity detection and convenient access to video material of multiple cameras and time instances. Even though there are continuous advances in this field, most approaches still suffer from high false positive rates or they are very specific to certain setups, e.g. abandoned bag detection [2] or traffic analysis [6]. Furthermore, recent advances in object tracking, crowd analysis, face recognition, and unusual event detection are not integrated into commercial systems since they are too complex to handle or compute, or since their output is too noisy for an automatic situation assessment.

Summing up, computer vision approaches in surveillance allow remarkably well results in certain disciplines, but the high-level classification “is everything alright?” has not been tackled yet. Besides the problem of imprecise knowledge about the actual condition of the scene, the nominal condition (the background knowledge) is also not present. This is crucial for detecting unusual events and surveillance in general, since critical events cannot be trained by example.





**Fig. 1.** System overview for a feature-based situation analysis system. For simplicity, the 3D tracking-by-recognition pipeline of only one camera is displayed. In a complete system, multiple tracking-by-recognition systems from different cameras are connected to one inference system.

In this paper we propose a solution to both problems. As displayed in Figure 1, we extract scene knowledge by the *tracking by recognition* approach. Since such a feature-based surveillance system is sufficiently reliable, we can use its output as facts in a reasoning system. Here, the actual condition is compared with a nominal condition. If both states differ in a critical way, a warning is generated in order to steer the operator’s attention. Such reasoning systems are widely used in medical applications and are thus manageable and reliable at the same time.

Our approach targets the automatic situation assessment for event-based video surveillance (ASEV) on airport aprons. However, since the methods used are quite general, they can be transferred easily to other scenarios. In the apron scenario, the conflict between privacy and safety is very high since ramp staff is monitored all the time and safety concerns are big. Since the whole approach can be applied without knowing the original images, we believe that privacy as well as safety can be enhanced.

This paper is organized as follows: In Section 2, we show how the image-based pipeline in object tracking can be replaced by a feature-based which enables learning object features by motion segmentation. By this, tracking by recognition can be used which is very robust and allows to deal with long-time occlusions. In Section 3, we show that on top of this, the 3D feature point cloud of an object can be learned, which is used for 3D tracking by recognition. In Section 4, we describe our reasoning system. In Section 5, we demonstrate how to create background masks for privacy protection and to direct the attention of the operators. Finally, a conclusion is given in Section 6.

## 2 Feature-Based vs. Image-Based Surveillance

### 2.1 Image-Based Surveillance

Traditional image-based approaches reason on a stream of camera images  $I_t(\mathbf{x})$ , where  $\mathbf{x}$  is a coordinate of a pixel in image  $I_t$  which may contain intensity, color, and depth information. To detect and track objects in a scene, the change in multiple images  $I_t$  is analyzed. A comprehensive survey can be found in [31]. In surveillance scenarios, two approaches are commonly used: background removal and optical flow.

In background removal, a binary foreground mask  $F(\mathbf{x})$  is defined for every  $\mathbf{x}$  by learning the probabilistic distribution  $p_b$  of the background

$$F(\mathbf{x}) = \begin{cases} 1 & p_b(I(\mathbf{x})|\mathbf{x}) < \tau_i \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In early approaches [41],  $p_b$  was modeled by a pixel-wise Gaussian  $\mathcal{N}(\mu(\mathbf{x}), C(\mathbf{x}))$  containing the two parameters standard deviation  $\mu(\mathbf{x})$ , which denotes an average background image, and variance, or covariance matrix respectively,  $C(\mathbf{x})$ , which describes the variability of pixel  $\mathbf{x}$  over time. More recent approaches use multivariate Gaussian distributions [39], non-linear colorspaces [34], and hierarchical modeling instead of pixel-wise [10]. In a post-processing step, obvious errors in  $F$  like very small or very elongated objects are deleted. Furthermore, special methods for shadow and reflection handling like [37] are applied. Since the background model itself is learned and updated using  $F$ , drifting occurs if the background is hidden by foreground objects for a long time or if foreground is falsely classified as background<sup>1</sup>. An example for this is if a foreground object is looking similar to the background (cf. Figure 2). On the other side, if the background is classified as foreground, it is not updated and the modeling becomes worse (cf. Figure 3). The main idea to circumvent this is to learn the background model from long time spans (one day or more) which has a very high computational complexity and which is not responsive if the background changes. In terms of artificial intelligence, the approaches suffer from the adaptivity vs. plasticity dilemma [8].

Optical flow approaches analyze the spatial difference between two consecutive images  $I_{t-1}$  and  $I_t$ , to find the discrete displacement field  $\mathbf{D}(\mathbf{x})$  by

$$\arg \min_{\mathbf{D}} \|I_{t-1}(\mathbf{x}) - I_t(\mathbf{x} - \mathbf{D}(\mathbf{x}))\|_2 . \quad (2)$$

---

<sup>1</sup> These approaches are *recursive* as the learning is performed on previous classifications. There also exist *non-recursive* background models which estimate  $p_b$  on the basis of  $N_t$  previous images, e.g. by computing the pixel-wise median [7]. Such approaches are not taken into account since  $N_t$  must be much larger than the amount of frames a foreground object may rest still. By this, the computational complexity becomes too high and the model loses responsiveness since an update would take  $N_t$  frames.



**Fig. 2.** Background removal using a multivariate background probability. Displayed are input image  $I_t$ , the average background  $\mu$ , and the foreground mask  $F$ . It can be observed, that the shirt of the left person is only partially detected as foreground since it looks similar to the background. Furthermore, shadows on the floor are detected as foreground.



(a) Original view at time  $t_0$ . (b) Illumination changes and reflections at  $t_0 + 10s$ . (c) Foreground from the background model of (b).

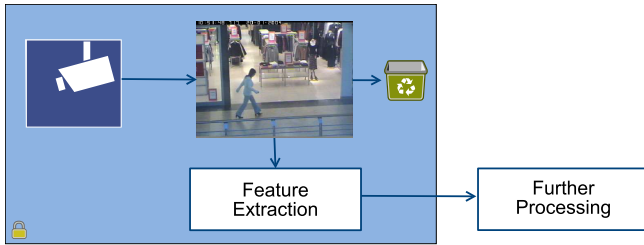
**Fig. 3.** Diverged Gaussian mixture background model caused by illumination changing faster than the model adapted

By assuming a static camera, the foreground  $F$  can be found from  $D$  by

$$F(\mathbf{x}) = \begin{cases} 1 & \|\mathbf{D}(\mathbf{x})\| > \tau_f \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

However, determining the optical flow in (2) is complex and since foreground objects do not necessarily move (e.g. a car waiting before traffic lights), this method can only be used as a prior for (1).

To recapitulate: In image-based methods only the background is described, as, in contrast to the foreground, it can be learned over a long time. However, the performance of the various approaches is still not sufficient for many real-world applications and high gains are still to be achieved [30]. A stable prior for motion segmentation is the optical flow which is very complex even for two consecutive images, but for reliable detection motion has to be analyzed over longer time spans. In the following section we propose to adapt the methods of image-based surveillance to feature-based. This has the advantage, that background as well as foreground can be learned and that motion can be analyzed over longer time spans.



**Fig. 4.** A feature-camera (blue frame) captures images, computes local features and only exports the features

## 2.2 Tracking in Feature-Based Surveillance

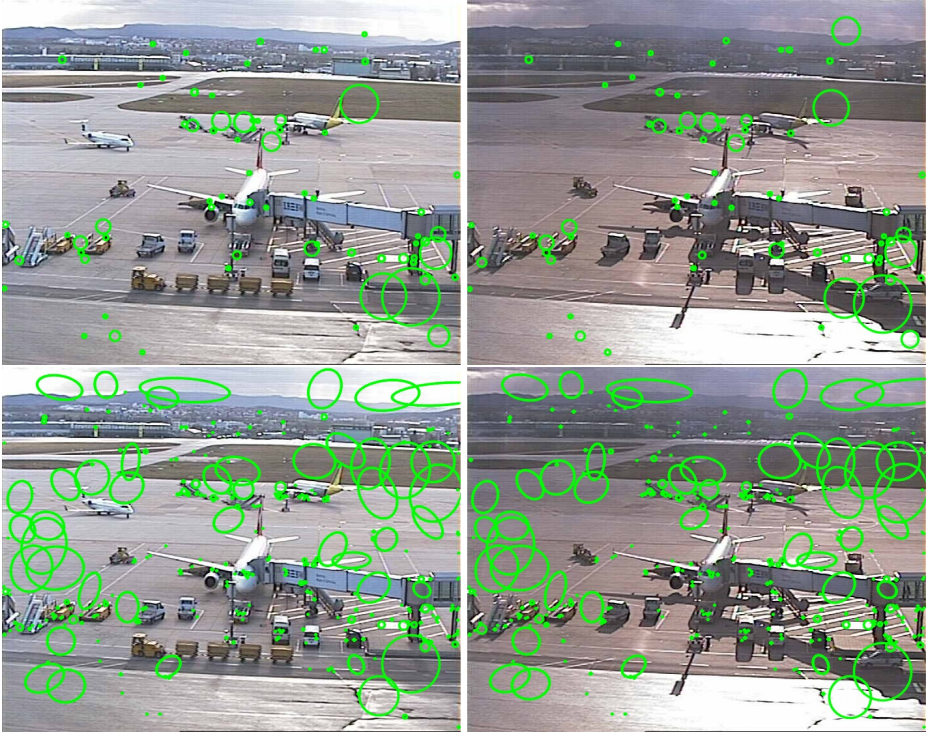
In the past decade, the combination of interest point detectors like SIFT [24] and Harris-affine [26] with local descriptor like SIFT, GLOH [27] and MSER [25] has been successfully applied in a high number of computer vision problems. The main reason for that is the fact, that establishing local image correspondences, which is one of the main computer vision problems, can be solved by inexpensive descriptor matching. Since local image descriptors are used to establish correspondences, such feature-based approaches are able to cope with partial occlusions and clutter. The descriptors are intentionally built such that changes in illumination as well as scaling and rotation of the image plane leaves them mostly unchanged. Thus, the main problem in modeling, namely that the background changes due to illumination, is suppressed up to a high degree when using local image descriptors. This can be observed in Figure 5, in which the illumination changes, which caused a background model to diverge in Figure 3, are still acceptable in order to establish correspondences between the two images. This gives hope that we can describe the background by means of features.

In the case of video surveillance, privacy protection plays an important role. Since for feature-based methods no original image data is needed, a *feature camera* could be used. As depicted in Figure 4, the features are extracted inside the camera such that no image data leaves the camera. By this, unauthorized access to the camera images becomes by far harder [2].

## 2.3 Learning Object Features by Motion Segmentation

In this section we demonstrate that by using local features, the background can be described even if the camera moves. Furthermore, we can also describe foreground objects and by this learn their local features. In contrast to image-based modeling, our feature-based approach distinguishes objects by their motions,

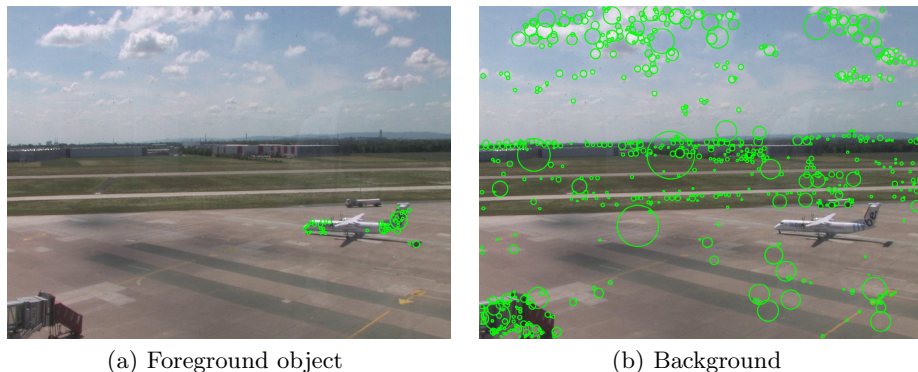
<sup>2</sup> Recently, methods to reconstruct images from local image features were proposed [40]. By this, the global scene layout could be recovered remarkably well. However, details cannot be not recovered with this method since they are mainly hallucinated.



**Fig. 5.** SIFT correspondences (top), and complementary NF feature [11] correspondences (bottom) between the views in Figures (a) and (b) which caused the image-based background model to diverge. From the point-of-view of illumination invariant features, the images are similar since the correspondences cover wide areas.

not their appearance. Thus, we extract the foreground by motion segmentation instead of building a pixel-wise foreground mask.

In the field of motion segmentation, feature trajectories  $T_i$  are clustered into groups of common motion. In the surveillance context, the camera is far from the object. By this, motion groups correspond to objects with different motions and motion segmentation becomes equivalent to object segmentation (cf. Figure 6). Motion segmentation approaches can be differentiated into subspace- and affinity-based approaches. Subspace-based approaches like [9,12,13,42] assume complete trajectories  $T_i$  which are inserted into a data matrix  $W$ . Since rigid object trajectories form linear subspaces in  $W$ , different object motions can be segmented by analysis of these subspaces. However, since we cannot provide complete data, we use an affinity-based approach like [5,17]. Here, the square affinity matrix  $A$  is computed which consists of pair-wise affinity measures  $a_{i,j}$  between trajectories  $T_i$  and  $T_j$ . In these measures, the spatial distance between  $T_i$  and  $T_j$  as well as their similarity in motion is included. In a final spectral clustering [28] step, the association of the trajectories to motion clusters is found.



**Fig. 6.** Motion segmentation of SIFT features. Although the airplane performs a turning operation in which perspective effects are non-negligible, *motion* segmentation corresponds to *object* segmentation.

Motion segmentation is applied to trajectories found from subsequent feature correspondences. In order to achieve longer trajectories, we apply the trajectory repair idea from [38]. The problem is computationally tractable since only a window over a time range sufficient for motion segmentation needs to be analyzed, here 5 s. Furthermore, the here-used independently-detected features allow satisfactory results when matching over a time span of 0.5 s, which is much longer than tracked features like KLT [36]. Thus, we analyze windows of  $N = 10$  frames taken at 2 Hz. By this, we can reliably segment motion if it is noticeably fast during the given window size and if the motion consists of enough features. The first constraint could be weakened by enlarging the window size. Regarding the second constraint, we deal with this by using high image resolution (up to 1.5 Mpel) or by using pan-tilt-zoom (PTZ) cameras scanning the scene with high zoom until they detect motion. To our experience, objects need to own approximately 10 to 15 detected features in order to reliably get detected (cf. Figure 7).

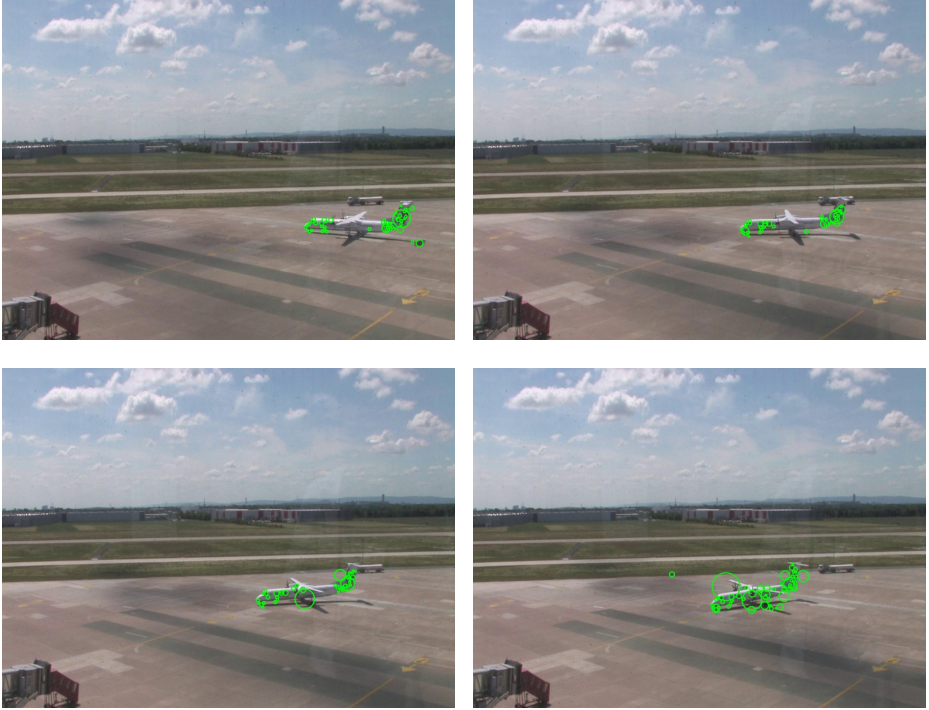
Motion segmentation extracts sets of local features  $\mathcal{M}_i(t)$ , corresponding to different motions  $i$  at frame  $t$ . We store these sets in the object feature database  $O = \{\mathcal{M}_1(t_1), \mathcal{M}_2(t_1), \dots, \mathcal{M}_1(t_2), \dots\}$  in order to compare the features with later input data. In contrast to image-based approaches, the background is treated as a regular object. As demonstrated in Figure 8, this allows using non-static cameras like PTZ cameras performing camera motion in the analyzed frames. Since the objects are described by their features and their geometric alignment, illumination changes as well as shadows and reflections do not pose major problems.



**Fig. 7.** Motion segmentation of small objects. Compared to the image resolution of  $1440 \times 1080 \text{ pel}^2$ , the objects are quite small with approximately  $110 \times 70 \text{ pel}^2$  (left) and  $100 \times 100 \text{ pel}^2$  (right). Similarly, the number of features (18 and 15, respectively) is only a fraction of the global scene (966 and 891, respectively).



**Fig. 8.** Motion segmentation under panning and tilting. The four images are taken during a time span of 2 s. As it can be observed, that although the segmented airplane is moving slowly compared to the panning, it is segmented correctly.



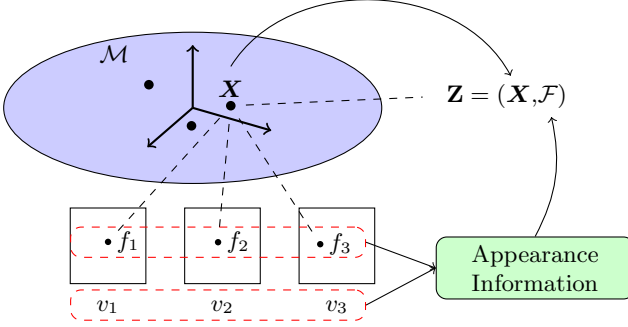
**Fig. 9.** Four views automatically learned from motion-segmented feature trajectories of the foreground object in Figure (a)

### 3 3D Object Learning and Recognition

The results of motion segmentation as described in Section 2.3 could be used for multiple instance learning (cf. Figure 9). However for critical applications such as airport surveillance, 2D object recognition is usually not sufficient for a fully-informed operation as it merely permits to find the 2D camera coordinates of the object location. In contrast, 3D tracking allows to recover complete 3D information, such as object location, pose and motion direction, expressing them with respect to a world coordinate frame. In these coordinates, safety rules of an airport apron can easily be expressed (cf. Section 4), e.g. the rule “only the scheduled airplanes may enter the taxi ways”.

One of the possible choices for the detection and tracking framework is the model-based approach pioneered in [18,32], where SIFT features [24] are used to reconstruct in an off-line fashion a 3D point cloud representing the target object. Once the model database has been assembled, on-line recognition and tracking can be performed by establishing putative correspondences between 3D model and current frame features and then estimating the 3D object pose.





**Fig. 10.** 3D feature  $\mathbf{Z}$  built from views  $v_1, v_2$ , and  $v_3$ . The 3D descriptor  $\mathcal{F}$  is established from the corresponding 2D descriptors  $f_1, f_2$ , and  $f_3$ , respectively.

### 3.1 3D Object Learning

The tracking-by-recognition approach requires first to build a set of 3D object models in an off-line stage. Here we start from a set of training views which are automatically found from motion-segmented trajectories  $T_i = (\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,n})$  (cf. Section 2.3 and Figure 6). So like in similar approaches [3,20,21,23,29], we detect and track SIFT features over visually close training views. The trajectories are input to a Structure from Motion (SfM) algorithm that outputs a 3D point cloud that represents the object structure. The feature descriptors  $f_{i,j}$ , observed at the respective 2D positions  $\mathbf{x}_{i,j}$  to which a corresponding 3D point  $X_i$  projects, are provided together with the 3D point coordinates. By doing so, it is possible to implement a 3D-2D feature matching at the recognition and tracking stage.

Since the set of descriptors can be highly redundant, particularly in case of long tracks, many of the above methods employ a feature quantization step. In [20], a hierarchical quantization is used for preserving matching ambiguities until the pose estimation step, where incoherent matches are dropped. Feature quantization can also be motivated by dimensionality reduction, as the 3D model size is usually too large to keep the system operating in real-time. This approach is shared by [3] and [21], where feature quantization is applied for outdoor scene reconstruction and image registration, respectively.

In order to form a 3D feature  $\mathbf{Z}$ , for each 3D point  $\mathbf{X}$  we compute a 3D feature descriptor  $\mathcal{F}$  containing appearance information from multiple views by applying a high-dimensional mean-shift clustering to the set of corresponding features:

$$\mathbf{Z} = (\mathbf{X}, \mathcal{F}). \quad (4)$$

In Figure 10, an example of building of a 3D feature  $\mathbf{Z}$  is shown. In this case,  $\mathcal{F}$  contains the matching 2D descriptors  $f_1, f_2, f_3$  from views  $v_1, v_2, v_3$ .

### 3.2 3D Object Recognition

Once the model database has been assembled, the general on-line operation envisages the creation of 2D-3D correspondences between frame features and model features, and the estimation of the pose by solving the projection problem.

A set of features is extracted in each frame and it is matched against the model feature set by using one of the following matching strategies. The most straightforward approach is to match the entire set of detected features against the model feature set by using a matching strategy based on the second-nearest-neighbor (2nn) distance ratio, as proposed by [24]. That is, a match  $(f, f_{nn})$  between a feature  $f$  and its nearest-neighbor feature  $f_{nn}$  is considered to be correct if

$$d(f, f_{nn}) < d(f, f_{2nn}) \cdot \tau, \quad (5)$$

where  $d(\cdot, \cdot)$  is an appropriate distance metric,  $f_{2nn}$  is the second nearest neighbor for  $f$  in the  $d(\cdot, \cdot)$  metric, and  $\tau$  is a threshold, given as 0.7 in the original paper [24]. Since the 2nn distance ratio strategy was conceived in order to reject false matches due to background clutter, it may remove many true positives if repetitive patterns or texture symmetries occur on the object surface. In [20] countermeasures are proposed based on dropping the 2nn strategy and employing hierarchical feature quantization and pose estimation constraints. Matches are created by thresholding their normalized cross correlation and stored along with their 3D location. Potentially spatial incoherent matches are kept until a pose estimation step, where geometric constraints will single out the true matches and discard the others. On the contrary, the 2nn approach can be maintained if difficult feature arrangements are handled by spatially clustering the original image feature set, e.g., by using mean shift clustering. Since features tend to cluster over the object surface, individual objects can be isolated before the matching step and thus, ambiguities can be avoided.<sup>3</sup> A visual example of the usefulness of spatial feature clustering is given in Figure 11.

Once feature clusters are established, object recognition and pose estimation is performed on the clusters, as represented in the block diagram given in Figure 12. Attention has to be paid as clustering may split or merge objects, visible in Figure 11.

After the matching, putative correspondences are established. Given a set of  $N$  2D-3D correspondences  $(\mathbf{x}_i, \mathbf{X}_i)$ , a projection matrix  $\mathbf{P}$  is to be computed such that

$$\mathbf{P} = \arg \min_{\mathbf{P}} \sum_{i=1}^N D(\mathbf{x}_i, \tilde{\mathbf{P}} \mathbf{X}_i)^2. \quad (6)$$

Thus,  $\mathbf{P}$  minimizes the sum of the squared re-projection errors  $D$  over all correspondences. Since the putative matches set contains outliers, a statistically robust approach is typically used in order to estimate the mathematical model

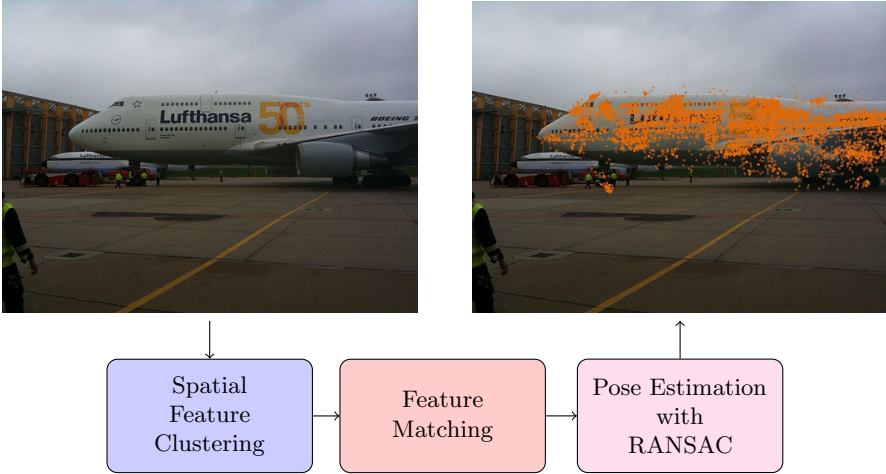
<sup>3</sup> Of course the motion segmentation methods from Section 2.3 could be applied here, too. However this clustering method only works if an object is currently moving. Thus, the tracking-by-recognition paradigm would be dismissed.



**Fig. 11.** By spatial feature clustering, the objects in the scene are segmented into five different clusters. This increases the inlier-outlier ratio for each cluster, and permits to avoid mismatches due to objects having a similar appearance, as in the case of the three airplanes.

underlying the samples. One of the most used algorithms is Random Sample Consensus (RANSAC) [14], in which a minimal subset of samples is iteratively used to estimate the model parameters, and the rest of the samples ranks the model consensus and can eventually be used to refine the parameters themselves.

If the minimal subset is created by randomly selecting the samples, no additional information regarding the importance of each sample and the relations among the samples themselves is used. Several approaches have been proposed in the literature in order to guide the RANSAC sampling by exploiting properties or constraints among the samples. E.g., in [20], a geometrical constraint based on the co-visibility of the 3D points in the sample set is used. After the first sample has been chosen, the part of the remaining samples that do not share any common view in the training images is discarded. This concept can be easily extended by giving the samples a weight-based priority computed from additional 3D information. For each feature, we compute weights on the basis on their frequency in the training images and of their co-visibility with other samples. These weights guide RANSAC towards a better selection, thus improving the final robustness and accuracy of the estimated pose. [23] only exploits feature priority in the matching step for the purpose of speeding up the process. Instead of using all model features for matching, they propose to use a subset of features selected on the basis of priorities representing feature frequency and co-frequency.



**Fig. 12.** On-line stage. SIFT features are detected and spatially clustered. Each cluster is matched against a database object and its pose is estimated using RANSAC. The model cloud is reprojected in orange to show the precision of the estimated pose.

In Table 1, an overview of the contribution of the guided sampling in terms of average iteration count for different inlier ratios is given until at least 75% of the inliers are found. The results are averaged over 1000 runs per frame for a short video sequence. It can be observed that our method is highly beneficial in real-time applications where the permitted number of iterations is small.

**Table 1.** Mean and standard deviation of the number of iterations for different inlier ratios

Inlier ratio	No weight	Guided Sampling
60%	$39.9 \pm 40.6$	<b><math>5.8 \pm 6.2</math></b>
50%	$110.5 \pm 113.3$	<b><math>9.4 \pm 12.6</math></b>
40%	$309.0 \pm 286.7$	<b><math>17.4 \pm 19.9</math></b>
30%	$627.4 \pm 515.0$	<b><math>19.0 \pm 27.6</math></b>
20%	$1428.5 \pm 1294.6$	<b><math>29.1 \pm 56.1</math></b>

After the minimal subset is determined, a method for estimating the pose  $\mathbf{P}$  given in Eq. (6) that best fits the 2D-3D point pairs is used. This is called the Perspective- $n$ -Point (PnP) problem. The algorithms for estimating the pose presented in the literature are countless, e.g., DLT, clamped DLT, POSIT, P4P, etc., and therefore the choice depends mainly on the complexity and time constraints given by the application considered. In case of real-time applications, like the one at hand, the Enhanced PnP (EPnP) method is a very common choice. It guarantees speed, as its complexity is only  $O(n)$ , and accuracy at the same time, as shown in detail in [22]. Finally, the estimated pose that holds the

maximum consensus among the entire set of correspondences is returned and thus, the object is considered detected.

### 3.3 3D Object Tracking

Regarding object tracking, different strategies are typically presented in the literature, as, e.g., tracking-by-recognition. Advantages of the later method are the absence of error drift and the fact that tracking failures do not affect successive frames as each frame is treated separately.

However, the appearance between the object model and its current projection on the image plane may vary too much. This can be due to the fact that the model was created off-line from a finite and small number of views and that SIFT features do not offer enough invariance. Therefore, it proves to be a hard problem if the object pose is far from the training images. In order to cope with this, [20] and [21] have proposed adding synthetic features created by deforming the training images in an affine way and extracting the features out of them. A clear disadvantage of this approach is the increase in size of the model, which can enlarge by more than one order of magnitude. Further, the distinctivity is lowered. A possible alternative is to use a model updating stage, where the model description is augmented after it has been detected. As a matter of fact, a matching image descriptor provides a reasonable approximation of the appearance of the same 3D point in the following frame. By this, the detection rate is boosted without a loss in precision. Some further images showing the tracking performance of our system are given in Figure 13.

## 4 Reasoning on Streams of Object Recognitions and Detections

The aim of the ASEV (automatic situation assessment for event-based video surveillance) system is to detect potentially safety-critical situations based on the image analysis results. To achieve this goal, the detected status is continually checked against safety rules, and violations are displayed as warnings to video surveillance operators. This section explains the challenges involved into this task, and how they were solved.

The reasoning component uses Semantic Web standards to represent the relevant expert knowledge. The airport domain is modeled using the Web Ontology Language (OWL Lite, [1]), including types and properties of objects found on the airport ramp, in particular the different vehicle types (cf. Figure 14). Safety rules cannot be expressed in OWL Lite, therefore this knowledge is captured as classical logical rules, represented in the Rule Interchange Format (RIF, [4]). Figure 15 shows as example a distance rule between moving planes and any other vehicle. These static expert knowledge is taken from official safety procedures, e.g. [15], from airport-internal guidelines, from work plans and flight schedules. During runtime, the facts describing the current situation are added to the knowledge



**Fig. 13.** Example of the 3D tracking of a toy plane. Different situations are presented: blank background, clutter and a combination of clutter and occlusion. The model cloud is reprojected in orange to show the preciseness of the estimated pose.

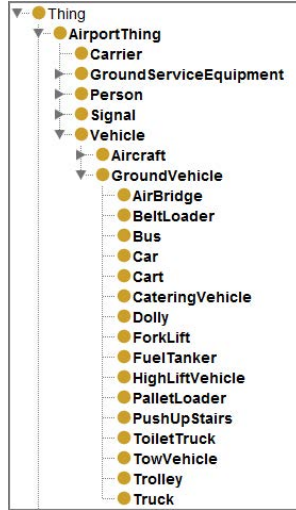


Fig. 14. Part of the airport ontology

base. These facts are generated by the object recognition algorithms described in Section 3.

Usual reasoning systems rely on the assumption that the knowledge based is rather static, while users pose a variety of queries over time. These systems are optimized to index and preprocess the facts and rules such that any arbitrary query can be processed efficiently. However, an update of the knowledge base invalidates intermediate results and requires a complete recomputation [33]. For the airport surveillance context, this assumption does not hold. New facts arrive every second, while only one query is ever posed to the system, i.e., “is there a safety critical situation?” In addition, the majority of incoming facts are not new, but fact updates concerning the position, orientation, and speed of planes and vehicles on the ramp. Therefore, existing reasoning engines could not be used to process the incoming object detection event stream efficiently.

Instead, we implemented a novel reasoning system, based on the Rete algorithm [16]. This algorithm works as follows: In an offline step, the domain knowledge captured in rules is converted into a directed graph, consisting of two types of nodes,  $\alpha$ - and  $\beta$ -nodes.  $\alpha$ -nodes represent conditions expressed in one of the rules, and  $\beta$ -nodes join these conditions. The leafs of this graph are *productions* which generate additional facts derived through the rule network. Figure 16 shows a part of the Rete network for the distance rules from Figure 15. Arriving facts are forwarded to all  $\alpha$ -nodes, which act as filters (shown on the top right). Matching facts are stored in the corresponding  $\alpha$  memory nodes. For example, the top  $\alpha$  memory node maintains a list of all objects of type *asev:Vehicle*. If a fact satisfies the constraint represented by an  $\alpha$ -node, it is forwarded to all  $\beta$ -nodes which rely on it. These nodes now perform a look-up in their  $\beta$  memory to check if there is a join possible. For example, the leftmost  $\beta$  join node matches objects which are vehicles and have a speed greater than 0. If a match could be

```

# Rule1: any vehicle with speed > 0 is moving
If And( rdf:type(?v asev:Vehicle) asev:speed(?v ?s) numeric-greater-than(?s, 0.0) )
Then Assert( asev:moving(?v) )

# Rule2: any aircraft with active anti-collision beacon is moving
If And( rdf:type(?a asev:Aircraft) asev:hasBeacon(?a ?acb) asev:active(?acb "true") )
Then Assert( asev:moving(?a) )

# Rule3: create warning if vehicle distance to moving aircraft is too low
If And(
  rdf:type(?a asev:Aircraft) asev:moving(?a)
  rdf:type(?b asev:Vehicle) asev:distance(?d ?a ?b)
  numeric-less-than(?d, asev:MinDistanceMoving) )
Then
  Assert(rdf:type(?w asev:DistanceWarning))
  Assert(rdfs:member(asev:warnings ?w))
  Assert(asev:participant(?w ?a))
  Assert(asev:participant(?w ?b))

```

Fig. 15. Distance rule between moving aircraft and vehicles, modeled in RIF

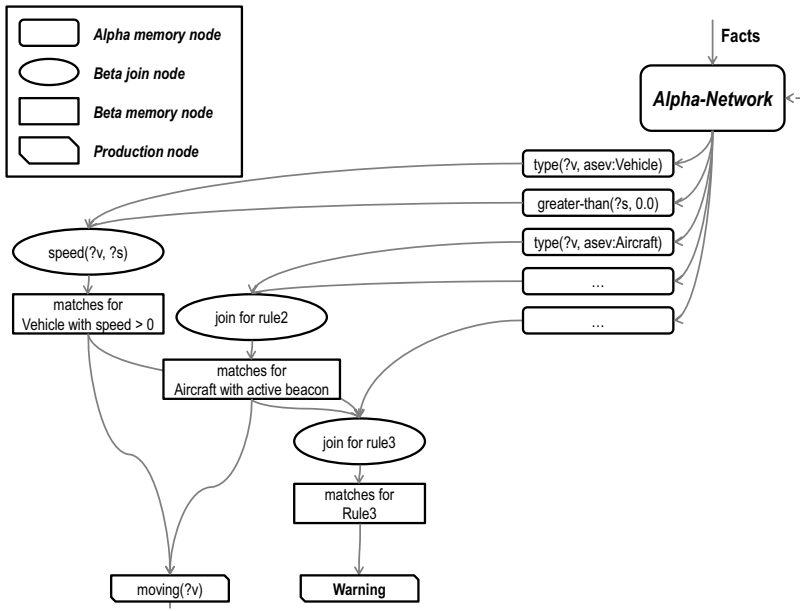


Fig. 16. Sample Rete network for distance rules from Figure 15

found, the result is forwarded to its successors. A successful join at a leaf node triggers a production, which creates a derived fact. These facts are fed back into the Rete-network to possibly derive further facts.



In the original Rete algorithm, nodes hold references to related facts. To optimize this approach for fact updates, we introduce an additional reverse index, which allows a lookup of all  $\alpha$ - and  $\beta$ -nodes maintaining this fact in their memory. When an update for a fact arrives, this enables very efficient updating of the respective node memories, to take the new value into account.

The reasoning engine is connected to the video analysis component via an XML event stream. The tracking-by-recognition component sends high-level object attributes such as type, position, speed, etc., and updates their values based on the analysis of each frame. If a safety-critical situation is detected, the reasoning engine sends a warning message to the video operator application.

## 5 Logging and Controlling Access to Surveillance Data

Let us imagine a feature-based system like the one presented here reports a critical event. An operator would then like to have a view on the scene before he takes further steps. If pure feature-cameras (cf. Figure 4) are used, this is not possible since no image signal leaves the camera. However, by introducing a system which controls access to images and logs this access, the use of surveillance image data becomes transparent. The following access rules are self-explaining:

- Since the scene content is known, it is logged which operator observes which object. Thus, mis-use by stalking is documented. Furthermore, regions with irrelevant information can be masked out (cf. Figure 17). In order to provide context to an operator, such regions may instead be faded out or blurred.
- An operator is allowed to get access to image data only if a critical event is detected. Overriding this is possible, but it is logged.

In order to quickly mask parts of the image, we use a method similar to the feature-based background removal from [35]. Given a set of features  $\mathcal{X}_+$  and  $\mathcal{X}_-$  which should be visible or not, respectively, we search for a binary segmentation  $s(\mathbf{x})$ , which is determined using spatial background and foreground probabilities  $p$ :

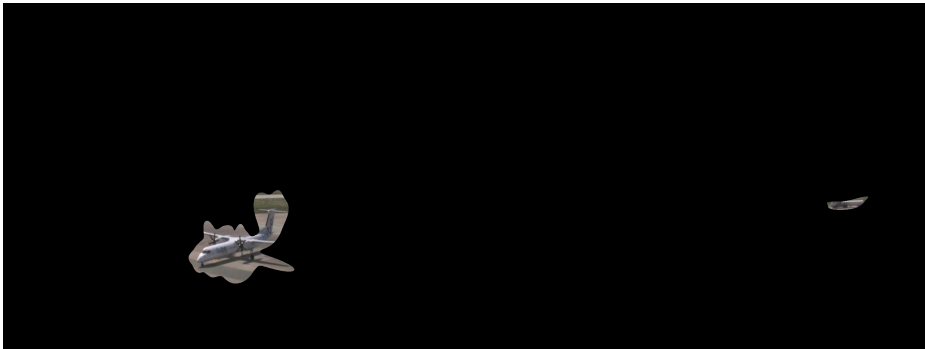
$$s(\mathbf{x}) = \begin{cases} 1 & p(\mathbf{x}|\mathcal{X}_+) > p(\mathbf{x}|\mathcal{X}_-) \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The spatial probabilities are estimated from kernel density estimation of  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  as

$$p(\mathbf{x}|\mathcal{X}) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mathbf{x}|\mathbf{x}_i, C_i) \quad (8)$$

using the normal kernel  $\mathcal{N}$  with adaptive bandwidth  $C_i$ , estimated from the covariance of the nearest 10% neighbors of  $\mathbf{x}_i$ .

By this, the attention of operators can be directed to important objects and unimportant image parts can be masked out. Furthermore, security and privacy is enhanced at the same time.



**Fig. 17.** Dense segmentation from SIFT feature density of the moving airplane and the resting fueling vehicle from the sequence displayed in Figure 9

## 6 Conclusion

In this paper, we have shown the concepts of the feature-based surveillance system ASEV (automatic situation assessment for event-driven video surveillance). It consists of a 3D tracking-by-detection system which inputs 3D information about visible objects into a reasoning system. By this, the current condition can be compared with a nominal condition which is specified by a rule set. Furthermore we have shown how to learn 3D models from motion and how foreground masks can be created from sets of foreground and background features.

Compared to traditional image-based surveillance, feature-based surveillance has the advantage that it is much more robust towards changes in illumination or background motion. Furthermore, our ASEV system allows to describe the foreground as well, which in turn enables tracking through long-time occlusions. The reasoning system facilitates comprehensive and reliable output event messages to operators. Since the scene interpretation can be used to mask out non-related scene content, the attention of surveillance operators is directed and privacy is enhanced at the same time.

**Acknowledgement.** Research was conducted inside the BMBF-funded project ASEV.

## References

1. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: Owl web ontology language reference. Tech. rep., World Wide Web Consortium (2009)
2. Bhargava, M., Chen, C.C., Ryoo, M.S., Aggarwal, J.K.: Detection of object abandonment using temporal logic. *Mach. Vis. Appl.*, 271–281 (2009)
3. Bhat, S., Berger, M.O., Sur, F.: Visual words for 3D Reconstruction and Pose Computation. In: *The First Joint 3DIM/3DPVT Conference* (March 2011)

4. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A., Reynolds, D.: Rif core dialect. Tech. rep., World Wide Web Consortium (2010)
5. Brox, T., Malik, J.: Object Segmentation by Long Term Analysis of Point Trajectories. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 282–295. Springer, Heidelberg (2010)
6. Buch, N., Velastin, S., Orwell, J.: A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems* 12(3), 920–939 (2011)
7. Calderara, S., Melli, R., Prati, A., Cucchiara, R.: Reliable background suppression for complex scenes. In: *Proc. ACM Video Surveillance and Sensor Networks (VSSN)*, pp. 211–214 (2006)
8. Carpenter, G.A., Grossberg, S.: The art of adaptive pattern recognition by a self-organizing neural network. *Computer* 21(3), 77–88 (1988)
9. Chen, G., Lerman, G.: Motion segmentation by scc on the hopkins 155 database. In: *Proc. ICCV Workshop on Dynamical Vision* (2009)
10. Chen, Y.T., Chen, C.S., Huang, C.R., Hung, Y.P.: Efficient hierarchical method for background subtraction. *Pattern Recogn.* 40, 2706–2715 (2007)
11. Dragon, R., Shoaib, M., Rosenhahn, B., Ostermann, J.: NF-Features – No-Feature-Features for Representing Non-textured Regions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 128–141. Springer, Heidelberg (2010)
12. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: *Proc. CVPR*, pp. 2790–2797 (2009)
13. Favaro, P., Vidal, R., Ravichandran, A.: A closed form solution to robust subspace estimation and clustering. In: *Proc. CVPR*, pp. 1801–1807 (2011)
14. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6), 381–395 (1981)
15. Flight Safety Foundation: Ground accident prevention ramp operational safety procedures. <http://flightsafety.org/archives-and-resources/ground-accident-prevention-gap>
16. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence* 19(1), 17–37 (1982)
17. Fradet, M., Robert, P., Perez, P.: Clustering point trajectories with various life-spans. In: *Proc. CVMP*, pp. 7–14 (2009)
18. Gordon, I., Lowe, D.G.: What and Where: 3D Object Recognition with Accurate Pose. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) *Toward Category-Level Object Recognition*. LNCS, vol. 4170, pp. 67–82. Springer, Heidelberg (2006)
19. Green, M.W.: The appropriate and effective use of security technologies in U.S. schools. Tech. rep., Sandia National Laboratories (September 1999)
20. Hsiao, E., Collet, A., Hebert, M.: Making Specific Features Less Discriminative to Improve Point-Based 3D Object Recognition. In: *CVPR*, pp. 2653–2660 (2010)
21. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From Structure-from-Motion Point Clouds to Fast Location Recognition. In: *CVPR*, pp. 2599–2606. IEEE (2009)
22. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An Accurate O(n) Solution to the PnP Problem. *IJCV* 81, 155–166 (2009)
23. Li, Y., Snavely, N., Huttenlocher, D.P.: Location Recognition Using Prioritized Feature Matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 791–804. Springer, Heidelberg (2010)
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *IJCV* 60, 91–110 (2004)

25. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* (2004)
26. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: *Proc. ICCV* (2002)
27. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence* 27(10), 1615–1630 (2005)
28. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: *Proc. NIPS*, pp. 849–856 (2001)
29. Park, Y., Lepetit, V., Woo, W.: Multiple 3D Object tracking for augmented reality. In: *ISMAR*, pp. 117–120 (September 2008)
30. Parks, D.H., Fel, S.S.: Evaluation of background subtraction algorithms with post-processing. In: *AVSS*, pp. 192–199 (2008)
31. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing* 14(3), 294–307 (2005)
32. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV* 66(3), 231–259 (2006)
33. Russell, S., Norvig, P.: *Inference in First-Order Logic*. In: *Artificial Intelligence: A Modern Approach*, ch. 9. Prentice Hall (2009)
34. Setiawan, N.A., Seok-Ju, H., Jang-Woon, K., Chil-Woo, L.: Gaussian Mixture Model in Improved HLS Color Space for Human Silhouette Extraction. In: Pan, Z., Cheok, D.A.D., Haller, M., Lau, R., Saito, H., Liang, R. (eds.) *ICAT 2006*. LNCS, vol. 4282, pp. 732–741. Springer, Heidelberg (2006)
35. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: *Proc. ICCV*, pp. 1219–1225 (2009)
36. Shi, J., Tomasi, C.: Good features to track. In: *Proc. CVPR*, pp. 593–600 (June 1994)
37. Shoaib, M., Dragon, R., Ostermann, J.: View-invariant fall detection for elderly in real home environment. In: *PSIVT* (November 2010)
38. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. *IJCV* 67(2), 189–210 (2006)
39. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *CVPR*, pp. 2246–2252 (1999)
40. Weinzaepfel, P., Jgou, H., Prez, P.: Reconstructing an image from its local descriptors. In: *CVPR* (2011)
41. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 780–785 (1997)
42. Yu, J., Chin, T.J., Suter, D.: A global optimization approach to robust multi-model fitting. In: *Proc. CVPR* (2011)

# Generalized Subgraph Preconditioners for Large-Scale Bundle Adjustment

Yong-Dian Jian, Doru C. Balcan, and Frank Dellaert

College of Computing, Georgia Institute of Technology  
ydjian@gatech.edu, {dbalcan,dellaert}@cc.gatech.edu

**Abstract.** We propose the Generalized Subgraph Preconditioners (GSP) to solve large-scale bundle adjustment problems efficiently. In contrast with previous work using either direct or iterative methods alone, GSP combines their advantages and is significantly faster on large datasets. Similar to [12], the main idea is to identify a sub-problem (subgraph) that can be solved efficiently by direct methods and use its solution to build a preconditioner for the conjugate gradient method. The difference is that GSP is more general and leads to more effective preconditioners. When applied to the “*bal*” datasets [2], our method shows promising results.

## 1 Introduction

Large-scale visual modeling with Structure from Motion (SfM) algorithms is an important problem. Recently, systems capable of handling millions of images have been built to realize this task [11,13,23], enabling automated 3D model generation from unstructured internet photo collections.

Bundle adjustment is used to find the optimal estimates of camera poses and 3-D points [26]. Mathematically speaking, it refers to the problem of minimizing the total reprojection error of the 3-D points in the images. The classical strategy to solve this problem is to apply a damped Newton’s method (e.g., Levenberg-Marquardt) and solve the reduced camera system by Cholesky factorization. However, this strategy does not scale well because the memory requirement of factorization methods grows quadratically with the number of variables in the worst case.

Several recent works suggest using iterative methods such as the conjugate gradient (CG) method to solve the linear systems arising in bundle adjustment, as its memory requirement grows only linearly with the number of variables. The convergence speed of the CG method depends on how *well conditioned* the original problem is [21]. Hence having a good preconditioner is crucial to make CG converge faster, yet most of the previous approaches [2,7,8,14] apply only standard preconditioning techniques, neglecting to exploit SfM-specific constraints.

In robotics, Dellaert et al. [12] proposed the Subgraph-Preconditioned Conjugate Gradients method (SPCG), which aims to combine the advantages of direct and iterative methods to solve 2-D Simultaneous Localization and Mapping (SLAM) problems. The main idea is to pick a subset of measurements that can be solved efficiently by direct methods, and use it to build a preconditioner for the CG method. They show that SPCG is superior to using either direct or iterative methods alone. However, for the

bundle adjustment problem, whose graph structure is bipartite and highly unbalanced, SPCG may over-estimate the uncertainty of the variables and hence lead to unsatisfactory preconditioners.

In this paper, we propose the Generalized Subgraph Preconditioners (GSP) that adapt SPCG to solve large-scale bundle adjustment efficiently [15]. While SPCG simply picks a subgraph of the *Jacobian* factor graph, GSP operates on the *Hessian* factor graph which is more general and leads to more effective preconditioners. From this perspective, the problem of designing a good subgraph preconditioner is reduced to picking a subset of the Hessian factors that (1) can be solved efficiently by direct methods, and also (2) make the linear systems well-conditioned.

An important open question in [12] is how to pick a good subgraph. To this end, we introduce the ideas developed in the field of combinatorial preconditioners to build good subgraph preconditioners [6]. The insight is that a good subgraph should not only be sparse but also have small structural distortion (*stretch*) with respect to the original graph. Yet finding the optimal subgraph that satisfies the above criteria is computationally intractable for large graphs. Instead we propose a greedy algorithm to construct a family of subgraphs by incrementally adding edges to reduce stretch without inducing large cliques in the factorization phase.

This paper has three contributions: we (1) adapt the ideas of SPCG to the bundle adjustment problem, (2) propose GSP which generalizes SPCG and leads to more effective subgraph preconditioners, and (3) develop a greedy algorithm based on the ideas in combinatorial preconditioners to construct a family of subgraph preconditioners. We use the proposed method to solve large-scale datasets and have promising results.

## 2 Bundle Adjustment

### 2.1 Formulation

Here we review the bundle adjustment, whose goal is to jointly estimate the optimal camera parameters and 3-D structure by minimizing the total reprojection error. We define  $\mathbf{X} = \{x_i\}_{i=1}^M$  as the camera parameters,  $\mathbf{L} = \{l_j\}_{j=1}^N$  as the 3-D points, and  $\mathbf{Z} = \{z_k\}_{k=1}^K$  as the measurements of the 3-D point  $l_{k_j}$  in camera  $x_{k_i}$ . We also define a function  $h_k(x_{k_i}, l_{k_j})$  that projects a 3-D point to an image (see Figure 1). The goal of bundle adjustment is to find the optimal cameras  $\mathbf{X}$  and 3-D points  $\mathbf{L}$  that minimizes the total reprojection error

$$\sum_{k=1}^K \|h_k(x_{k_i}, l_{k_j}) - z_k\|^2. \quad (1)$$

Equation (1) is nonlinear and has no closed-form solution, but suppose we know some initial estimates of the cameras parameters and 3-D points, we can apply the first-order Taylor expansion to linearize Equation (1) as

$$\sum_{k=1}^K [h(x_{k_i}, l_{k_j}) + \frac{\partial h(x_{k_i}, l_{k_j})}{\partial x_{k_i}} \delta x_{k_i} + \frac{\partial h(x_{k_i}, l_{k_j})}{\partial l_{k_j}} \delta l_{k_j} - z_k]. \quad (2)$$

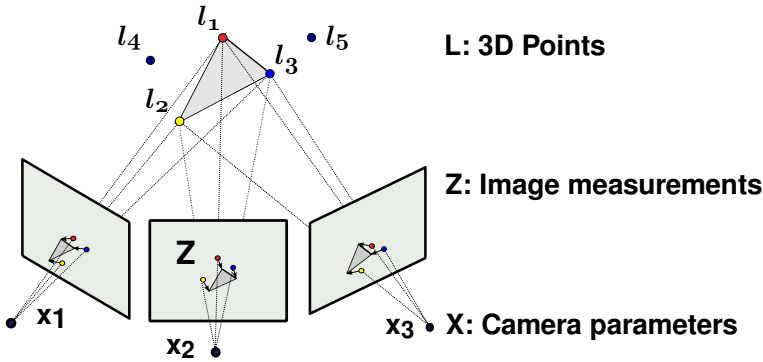


Fig. 1. The bundle adjustment problem

By setting the first-order derivative of the measurements in Equation (2) to zero, we can build a linear system

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{b}, \quad (3)$$

where  $\mathbf{A}$  is a sparse rectangular matrix containing the Jacobian of the measurements with respect to the cameras and 3-D points,  $\boldsymbol{\theta}$  is a vector that concatenates all  $\delta x_i$  and  $\delta l_j$ , and  $\mathbf{b}$  is a vector that concatenates the negative measurement errors. Then we solve Equation (3) and use its solution to update the current estimates. This process is repeated until convergence. We can see that solving bundle adjustment is equivalent to solving a sequence of linear systems. An alternative to the second step is to form and solve the *normal equation*

$$(\mathbf{A}^T \mathbf{A})\boldsymbol{\theta} = \mathbf{A}^T \mathbf{b}, \quad (4)$$

where  $\mathbf{A}^T \mathbf{A} \approx \mathbf{H}$  is a first-order approximation to the Hessian of the total reprojection error in Equation (1). Unfortunately, this method may not converge to the local minimum if the initial estimate is close to a saddle point. To to resolve this problem, one can solve a regularized linear system

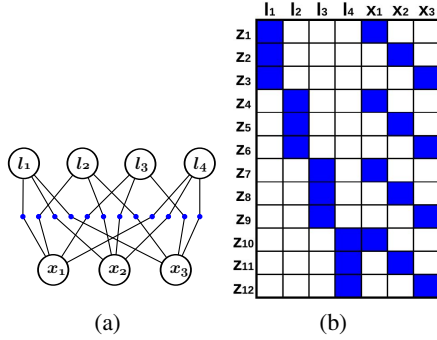
$$(\mathbf{A}^T \mathbf{A} + \lambda \mathbf{D})\boldsymbol{\theta} = \mathbf{A}^T \mathbf{b}, \quad (5)$$

where  $\lambda$  is a non-negative scalar, and  $\mathbf{D}$  can be an identity matrix or the diagonal of  $\mathbf{A}^T \mathbf{A}$ . In bundle adjustment, the Levenberg-Marquardt algorithm is used to update the value of  $\lambda$  according to quality of the solution. Note that the least-square linear system corresponding to the normal equation (5) is

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda \mathbf{D}} \end{bmatrix} \boldsymbol{\theta} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \quad (6)$$

## 2.2 Jacobian Factor Graph Representation

The bundle adjustment problem can also be considered as an inference problem on a factor graph. In particular, the sparse Jacobian matrix  $\mathbf{A}$  in Equation (3) can be regarded



**Fig. 2.** A toy bundle adjustment problem with three cameras and four 3-D points. All of the 3-D points are observed by all of the cameras. (a) The Jacobian factor graph. The vertices denote the camera and the 3-D point variables. The blue dots are the factors, and each factor indicates the squared error term of a projection measurement. (b) The symbolic representation of the Jacobian matrix  $\mathbf{A}$ . Each row denotes one Jacobian factor, and each column indicates one variable.

as a *Jacobian* factor graph, where the vertices are the cameras and the 3-D points, and each factor denotes the squared error term (block row) of a measurement. Figure 2 illustrates the idea with a simple example. Suppose we define the likelihood of a factor as an exponential function of the negative squared error

$$P(z_k|x_{ki}, l_{kj}) \propto \exp\left\{-\frac{\|h(x_{ki}, l_{kj}) - z_k\|^2}{2\sigma^2}\right\}, \tag{7}$$

we can see that the maximum likelihood estimator of the factor graph is the minimizer of Equation (1), i.e.

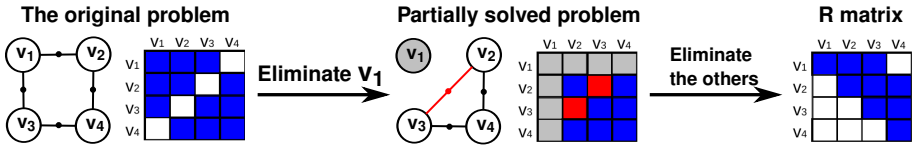
$$\operatorname{argmax}_{\mathbf{X}, \mathbf{L}} \prod_{k=1}^K P(z_k|x_{ki}, l_{kj}) = \operatorname{argmin}_{\mathbf{X}, \mathbf{L}} \sum_{k=1}^K \|h(x_{ki}, l_{kj}) - z_k\|^2 \tag{8}$$

This connection provides a foundation to the subgraph preconditioners.

### 2.3 Direct Methods

There are two ways to solve linear systems and the first one is called direct methods. They work by factorizing the matrix to the product of an upper triangular matrix  $\mathbf{R}$  and its transpose, followed by a backward and forward substitution step. For instance, we can use QR factorization to solve the linear least-square problem in Equation (3), and use Cholesky factorization to solve the normal equation in Equation (4) [25]. On factor graph, direct methods can be explained as a sequence of variable eliminations. Each time we eliminate a variable (vertex), we will instantiate a new factor connecting to all of its neighbors. After eliminating all of the variables, we will get an upper triangular matrix  $\mathbf{R}$  as a result. The process is illustrated in Figure 3. The variable elimination ordering is very important to the efficiency of direct methods. Using a good ordering





**Fig. 3.** An illustration of how direct methods factorize an  $\mathbf{H}$  matrix into  $\mathbf{R}^T \mathbf{R}$ . Suppose we have a variable elimination ordering. On the factor graph, each time we eliminate a vertex (variable), we will introduce a new factor (red) connecting to all of its neighbors. After eliminating all of the vertices, we will get the factorized matrix  $\mathbf{R}$ .



**Fig. 4.** An illustration of how the elimination ordering affects the sparsity of  $\mathbf{R}$  matrix. Suppose we eliminate the leaf vertices first, the  $\mathbf{R}$  matrix will be very sparse. Yet if we eliminate the center vertex first, it will introduce a clique over the remaining vertices, and hence the  $\mathbf{R}$  matrix becomes very dense, which negatively affects the performance.

will result in a sparse  $\mathbf{R}$  matrix and make the forward and backward substitutions more efficient. Figure 4 shows how the ordering affects the sparsity of the factorized matrix.

Using direct methods to solve bundle adjustment has been well-studied in the literature [16][17][26]. The common practice is to eliminate all 3-D points first, and use Cholesky factorization to solve the reduced camera system. Yet as shown in [28][14], this strategy only works well for small problems, but does not scale satisfactorily because (1) the cost of forming and storing the reduced camera systems is prohibitive for large problems, and (2) building the reduced camera system could destroy the sparse problem structure and hence make it harder to solve. Therefore direct methods cannot be directly applied to solve large-scale bundle adjustment without using hierarchical or incremental techniques [19][22].

### 2.4 Iterative Methods

The second way to solve linear systems is called iterative methods. They are better than direct methods for large problems because they involve only simple operations and require less memory, but they may suffer from slow convergence if the original problem is *ill-conditioned*.

The conjugate gradient (CG) method is the most efficient variant of iterative methods, but the convergence speed still depends on the condition number of the linear system, which is defined as the ratio of extreme eigenvalues of the matrix  $\mathbf{A}^T \mathbf{A}$ .

Several preconditioning techniques have been applied to make bundle adjustment well-conditioned. Agarwal et al. [2] examined the performance of several standard preconditioners and implementation strategies on large-scale datasets. Byröd and Åström

[7][8] proposed to use multi-scale and the block Jacobi preconditioners respectively. Jeong et al. [14] suggested using the band-diagonal of the reduced camera system as a preconditioner. Yet these methods are very generic: We show that by exploiting the problem structure of bundle adjustment we can obtain better preconditioners.

### 3 Combining the Best of Direct and Iterative Methods

#### 3.1 Variable Reparameterization and Preconditioning

Re-parameterizing the variables can result in faster convergence for iterative methods. In the robot mapping and localization problem, Olson et al. [20] showed that if the robot poses are parameterized in the global coordinate system, it takes a long time to propagate the loop closure constraints through the graph, but suppose the robot poses are incrementally parameterized along the odometry chain, so that the new variables denote the difference between two consecutive poses, they show that it makes the stochastic gradient descent method converge faster. Generally speaking, this re-reparameterization can be considered as a linear transformation  $\mathbf{R}$  between two domains.

Similarly, the preconditioned conjugate gradient method [21] also uses a preconditioner  $\mathbf{R}$  to linearly re-parameterize the problem such that the condition number becomes smaller and it can converge faster. This point of view indicates that linear variable re-parametrization is essentially a preconditioning process.

#### 3.2 Subgraph-Preconditioned Conjugate Gradient Method

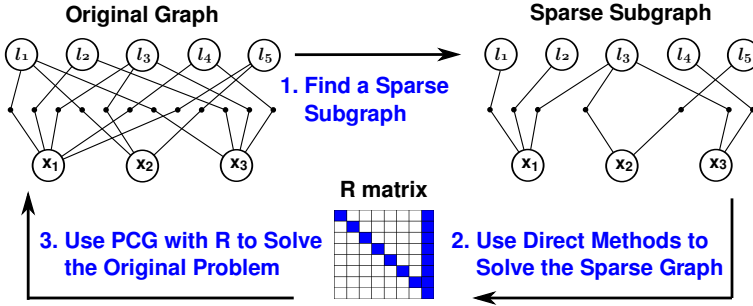
Dellaert et al. [12] proposed the Subgraph-Preconditioned Conjugate Gradient (SPCG) method, which aims to combine the advantages of direct and iterative methods to solve 2-D Simultaneous Localization and Mapping (SLAM) problems. The main idea is to identify a sub-problem (subgraph) that can be solved efficiently by direct methods (e.g., a subgraph with small tree-width) and use it to build a preconditioner for the conjugate gradient method. They show that this technique is a better alternative to using either direct or iterative methods alone. Figure 5 illustrates the key steps of the algorithm.

Here we show how SPCG works in detail. Suppose we want to solve a linear system (Jacobian factor graph) as in Equation (6). We pick a subset of the rows (factors), and denote it as  $(\mathbf{A}_1, \mathbf{b}_1)$ , and denote the remaining rows as  $(\mathbf{A}_2, \mathbf{b}_2)$ . We can re-arrange the linear system in Equation (6) as

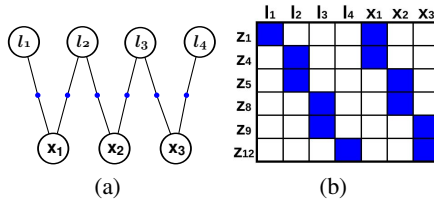
$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \boldsymbol{\theta} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \quad (9)$$

After applying QR factorization to  $\mathbf{A}_1$ , we have  $\mathbf{A}_1 = \mathbf{Q}_1 \mathbf{R}_1$ . By left-multiplying the upper part with  $\mathbf{Q}_1^T$ , we get

$$\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{A}_2 \end{bmatrix} \boldsymbol{\theta} = \begin{bmatrix} \mathbf{Q}_1^T \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \quad (10)$$



**Fig. 5.** An illustration of the SPCG method. Suppose on the left is the original factor graph. SPCG has three main steps: (1) Pick a sparse subgraph out of the original one. (2) Use direct methods to factorize this sparse subgraph. This step is efficient because a good variable elimination ordering for a sparse graph is always available. (3) Use the  $\mathbf{R}$  matrix of the subgraph as the preconditioner in the preconditioned conjugate gradient method to solve the original problem.



**Fig. 6.** An example that illustrates the SPCG technique. (a) The Jacobian factor graph that corresponds to a subset of the measurements (sub-problem) in Figure 2 (b) The symbolic matrix representation of the subgraph.

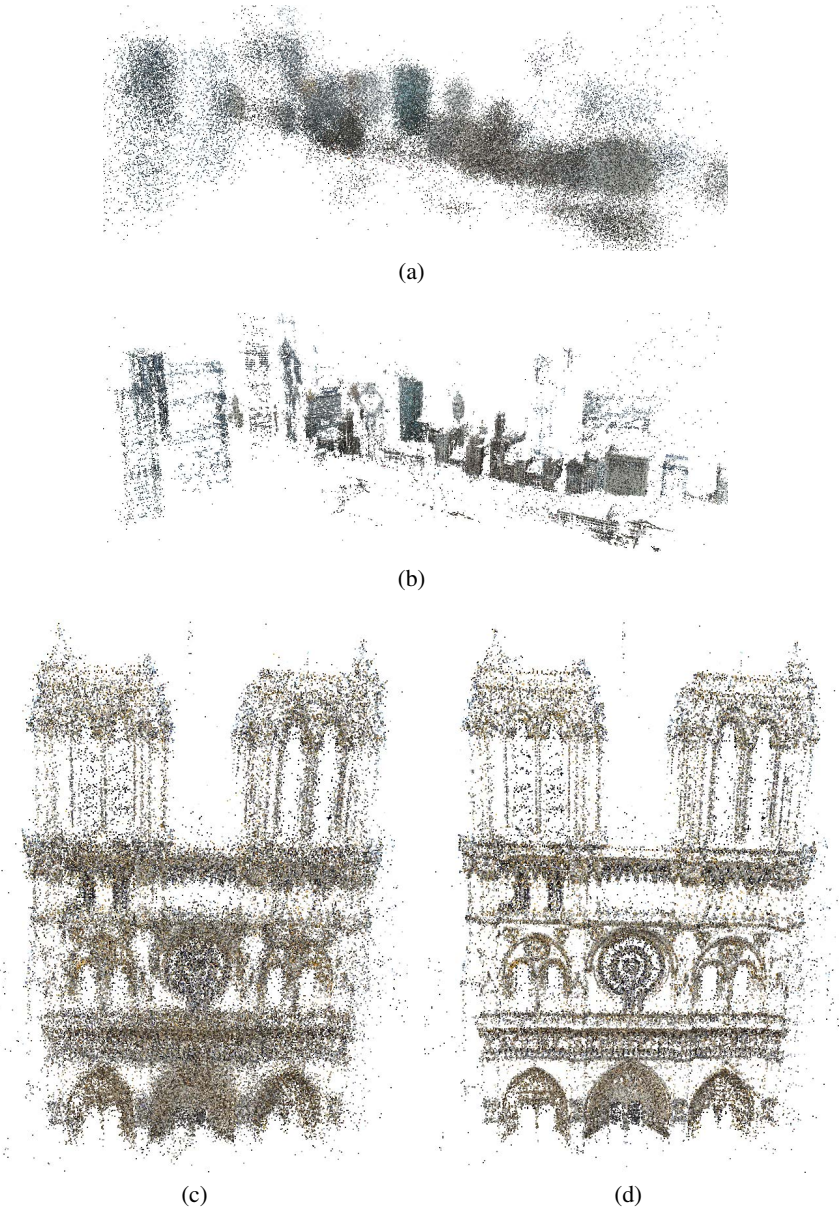
Suppose  $\mathbf{c}_1 = \mathbf{Q}_1^T \mathbf{b}_1$  and  $\bar{\boldsymbol{\theta}} = \mathbf{R}_1^{-1} \mathbf{c}_1$  is the optimal solution by considering only the upper part of Equation (9). Then by re-parameterizing  $\mathbf{y} = \mathbf{R}_1(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})$ , we have

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{A}_2 \mathbf{R}_1^{-1} \end{bmatrix} \mathbf{y} = \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_2 \end{bmatrix}, \tag{11}$$

where  $\mathbf{c}_2 = \mathbf{b}_2 - \mathbf{A}_2 \mathbf{R}_1^{-1} \mathbf{c}_1$ . Equation (11) couples the solution of the subgraph part ( $\mathbf{R}_1^{-1}$ ) to precondition the remaining part. The intuition behind the re-parameterization is that we penalize the deviation of  $\mathbf{y}$  from the subgraph solution  $\bar{\boldsymbol{\theta}}$ . Finally Equation (11) is solved by using the least-squares variant of the conjugate gradient method [4].

Figure 6 illustrates the SPCG technique with an example. Suppose we pick a spanning tree of the original graph as in Figures 6(a) and 6(b). We can use direct methods to factorize the spanning tree efficiently. Then we use the factorized matrix to precondition (re-parameterize) the original problem.

In addition, we also visualize the solutions obtained from the subgraph and the solutions from the original graph in Figure 7. We can see that although the solution of the subgraph is blurry and hence inferior to that of the original graph, we can use it to build a preconditioner to solve the original graph efficiently.



**Fig. 7.** The solutions obtained from solving (a)(c) the subgraph and (b)(d) the original graph on the *Chicago-2* dataset (from Grant Schindler) and the *Notre-Dame* datasets [23] respectively. Note that the solutions of the subgraphs are more blurry than (inferior to) those of the original graphs, but they could serve as good preconditioners to solve the original graph.

## 4 Generalized Subgraph Preconditioners

Although SPCG works well for 2-D pose SLAM problems, its performance is actually worse than the Jacobi preconditioner, a simple and empirically effective preconditioner [28][14], in our experiments on large-scale bundle adjustment. This indicates that we need a different representation to design subgraph preconditioners.

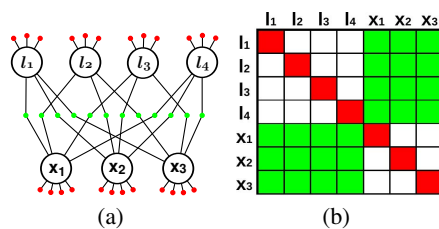
To this end, we propose the Generalized Subgraph Preconditioners (GSP), which generalize SPCG and are more suitable for large-scale bundle adjustment. While SPCG works on the *Jacobian* factor graph where each measurement corresponds to a Jacobian factor, GSP works on the *Hessian* factor graph where each measurement contributes three factors to the graph. We will show that this finer-grained graph possesses greater representation power than the Jacobian factor graph.

Compared to conventional matrix preconditioning machinery, GSP not only provides an expressive language to design subgraph preconditioners, but also explains the standard Jacobi preconditioner naturally.

### 4.1 Hessian Factor Graph Representation

To gain insight into the performance properties of both Jacobi and SPCG preconditioners, we investigate the structure of the *Hessian* matrix  $\mathbf{H} \approx \mathbf{A}^T \mathbf{A}$  appearing in the normal equation (4). The Hessian matrix can also be represented as a graph, more specifically a Gaussian Markov Random Field (GMRF). Every principal sub-matrix of  $\mathbf{H}$  corresponds to the information matrix of the conditional distribution given the other variables [11][18]. In this sense, solving the GMRF is analogous to solving Equation (4).

Yet a GMRF is usually represented as an undirected graph which is not expressive enough for designing subgraph preconditioners. It prompts us to resort to a finer-grained Hessian factor graph representation. The main difference is that we create two unary and one binary factors out of each measurement, and accumulate all of them in the *Hessian*



**Fig. 8.** The Hessian representation of the bundle adjustment problem in Figure 2. (a) The Hessian factor graph. The red dots denote unary factors while the green dots denote binary factors. This representation resembles to the Gaussian Markov Random Field representation [11][18]. (b) The symbolic representation of the Hessian matrix  $\mathbf{H} \approx \mathbf{A}^T \mathbf{A}$ . Both rows and columns indicate variables. A diagonal (red) block indicates the certainty of a variable given the other variables are known. An off-diagonal block indicates whether two variables are correlated given that the other variables are known. Each non-zero off-diagonal (green) block corresponds to a Jacobian factor in Figure 2(a) or a binary Hessian factor in (a).

factor graph. The number of unary factors attached to a variable is equal to the number of the associated measurements, with one binary factor per measurement.

As an example, consider the measurement between  $x_0$  and  $l_0$  in Figure 2(a) and assume  $A_{x_0}$  and  $A_{l_0}$  are the corresponding block entries in the first row of the Jacobian matrix in Figure 2(b). Since the Hessian matrix is the sum of outer product of the block rows of the Jacobian matrix, we can see that this measurement actually corresponds to three terms in the Hessian matrix:  $A_{x_0}^T A_{x_0}$ ,  $A_{l_0}^T A_{l_0}$  and  $A_{x_0}^T A_{l_0}$ . Notice that the first two are unary factors of  $x_0$  and  $l_0$ , and the third is a binary factor between them. They encode the information contributed by this measurement to the conditional Gaussian densities. Repeating this process for all measurements, we can build the Hessian factor graph representation illustrated in Figure 8(a).

From this perspective, the problem of designing a good subgraph preconditioner is reduced to picking a subset of Hessian factors from the graph that (1) can be solved efficiently by direct methods, and also (2) make the linear systems well-conditioned. Once a subgraph is selected, we can use sparse direct methods to factorize the linear system (i.e.,  $\mathbf{H}_1 = \mathbf{R}_1^T \mathbf{R}_1$ ) and use  $\mathbf{R}_1$  as the preconditioner in the conjugate gradient method. The detail of how to pick a subgraph will be discussed in Section 5.

GSP is more expressive than SPCG because we can always build a Hessian factor graph from a subset of measurements, but not vice versa. For instance, suppose we want to construct a Hessian factor subgraph as in Figure 9 by picking a subset of measurements. One can see that no subset of Jacobian factors in Figure 2(a) corresponds to this Hessian factor graph. Hence the GSP is indeed a generalization of SPCG.

The difference between GSP and SPCG is critical for large-scale bundle adjustment, whose graph structure is bipartite and highly unbalanced. The amount of information that SPCG brings in for each variable corresponds to the associated measurements in the subgraph. In bundle adjustment, if SPCG picks a spanning tree as the subgraph, then it can only collect at most two out of potentially thousands of unary factors for the camera vertices. This results in over-estimating the uncertainty of the variables and hence leads to unsatisfactory preconditioners. This idea is illustrated in Figure 10. Adding more measurements to the subgraph might help, but it also makes it harder for direct methods to solve the subgraphs. In contrast, GSP provides the flexibility to keep part or all of the unary factors (information) for each variable, and hence overcomes this problem.

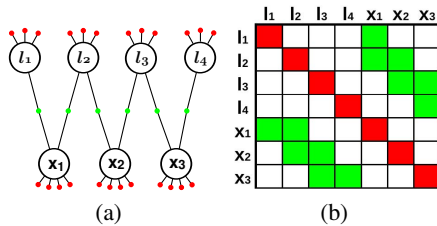
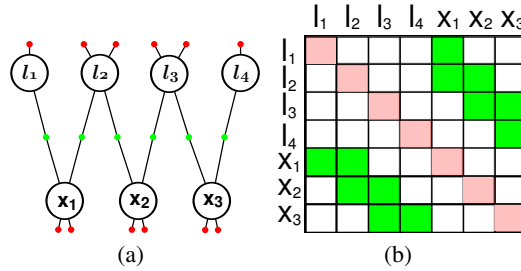


Fig. 9. A subgraph that GSP can generate but SPCG cannot

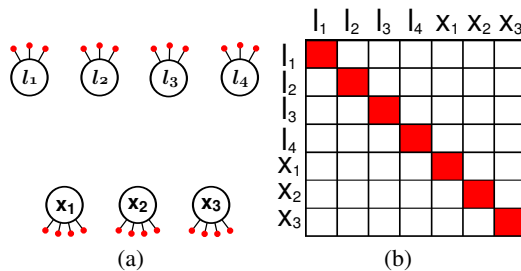


**Fig. 10.** The Hessian representation of the sub-problem in Figure 6. (a) The Hessian factor graph with the corresponding unary and binary factors. (b) The symbolic matrix of the sub-problem. The non-zero off-diagonal blocks are identical to those in Figure 8(b), but the diagonal entries are smaller than those in Figure 8(b). It leads to over-estimating the uncertainty of the variables, especially for the camera variables. This is problematic for large-scale bundle adjustment where the graph is bipartite and unbalanced.

### 4.2 The Jacobi Preconditioner

The Jacobi preconditioner is a generic technique and it has been shown empirically effective for large bundle adjustment [2,8,14]. Here we show that the Jacobi preconditioner has a simple explanation within the GSP framework. The Jacobi preconditioner works by taking only the diagonal entries of the Hessian matrix, and discarding all off-diagonal entries [21]. A simple generalization is the block Jacobi preconditioner which treats each camera and each 3-D point as an entity, and it corresponds to picking the block diagonal of the Hessian matrix. The block Jacobi preconditioners can be solved efficiently because all blocks are independent.

In the GSP machinery, the block Jacobi preconditioner corresponds to picking all of the unary factors and discarding all of the binary factors of in the Hessian factor graph. The idea is illustrated in Figure 11. Note that hereafter when we refer to the Jacobi preconditioner, we actually mean the block Jacobi preconditioner.



**Fig. 11.** Block Jacobi preconditioner of the toy problem

## 5 The GSP- $n$ Preconditioners

### 5.1 Matrix Preconditioners

Conventional matrix preconditioning techniques focus more on the efficiency of solving the preconditioners rather than on directly minimizing the condition number of the preconditioned system [21]. For example, the Jacobi preconditioner offers good computational efficiency by discarding the conditional correlation between variables. The incomplete Cholesky preconditioner controls the computational cost by limiting the amount of fill-in and discarding negligible entries during the factorization process. Although these techniques work to some extent in practice, deriving theoretical bounds on their condition numbers is generally non-trivial, and their actual meaning is also hard to interpret graphically or probabilistically.

### 5.2 Combinatorial Preconditioners

Recently, combinatorial (graph) preconditioners have been studied to analyze and construct effective preconditioners for the conjugate gradient method. Promising results have been reported on solving linear systems with symmetric and diagonally dominant matrices [6,24]. The main idea is to find ultra-sparsifiers such that the original graph and the approximating graph have similar conductance – a measure of how fast information travels between different parts of the graph. Insisting on sparse approximating graphs produces preconditioners that can be solved efficiently by direct methods, while maintaining the graph conductance effectively reduces the condition number of the preconditioned systems, therefore the number of CG iterations.

If the subgraph is restricted to be a spanning tree, Boman and Hendrickson [6] recognized that the condition number of the preconditioned system is upper bounded by the *stretch* of the original graph with respect to the spanning tree. More specifically, suppose  $G = (V, E, w)$  is the graph of the original system where  $V$ ,  $E$  and  $w$  denote the vertices, edges and the weights of the edges respectively. If  $T$  is a spanning tree of  $G$ , then for every edge  $e = (u, v) \in E$ , there is a unique path in  $T$  connecting  $u$  and  $v$ . The stretch of  $e$  with respect to  $T$  is defined as

$$\text{st}(T, e) = \sum_{f \in P(T, e)} \frac{w(e)}{w(f)}, \text{ for } e \in E \quad (12)$$

where  $P(T, e)$  denotes the edges on the unique path between  $u$  and  $v$  in  $T$ . The stretch of  $G$  with respect to  $T$  is defined as the sum of the stretches of all the edges in  $G$ :

$$\text{st}(T, G) = \sum_{e \in E} \text{st}(T, e). \quad (13)$$

Intuitively speaking, the higher the stretch of a tree, the more time it takes for information to percolate, negatively affecting convergence.

If we relax the restriction and consider a general subgraph, a common practice is to use a *low-stretch* spanning tree as a skeleton and augment it with additional edges to further reduce the stretch. However, when additional edges are added to the subgraph,



not only may the subgraph take longer to build, but also the preconditioners will become more expensive to apply in the conjugate gradient method. Clearly, there is a trade-off between the quality of the preconditioner and the time required to build and apply it.

### 5.3 The GSP- $n$ Preconditioners

Finding the optimal subgraph is computationally intractable for large problems. Instead we propose a greedy algorithm to construct a family of subgraphs with adjustable complexity. On top of these subgraphs, we use GSP to build subgraph preconditioners. The resulting preconditioners are called the GSP- $n$  preconditioners, where  $n$  is a parameter that controls the complexity of the subgraph.

The bundle adjustment graph is a bipartite graph  $G = (X, L, E)$ , where  $X$  denote the camera and  $L$  denote the 3-D points vertices on the two sides of  $G$ . Each edge in  $E$  denotes a measurement that connects the corresponding camera and point vertices.

The goal is to find a subset  $E_S$  of  $E$ , such that (1) the resulting subgraph  $G_S$  has low stretch with respect to  $G$ , and (2) the maximum size of the induced cliques does not exceed the predefined parameter  $n$ . By the maximum size of the induced cliques we actually mean the clique number in the factorization phase, which can indirectly affect the computational complexity. A straightforward strategy would be to use a low-stretch spanning tree of  $G$  as the subgraph, but this strategy is sub-optimal because it does not exploit the bipartite and unbalanced nature of  $G$ .

Here we introduce some notation to facilitate the explanation. We denote  $X(l)$  as the set of cameras associated with a 3-D point  $l$ , and  $E(l)$  as the corresponding set of edges (measurements). Note that by picking  $t$  edges from  $E(l)$  into the subgraph, we will induce a clique of size  $t$  between the corresponding cameras after eliminating the 3-D point  $l$  in the factorization phase. Moreover, if the edges and the elimination ordering are not chosen appropriately, even larger cliques will appear in the factorization phase.

Here we describe a greedy algorithm to construct a family of subgraphs. First, we build a camera graph  $G_X$  where the vertices consist of all cameras and the edge weight between two cameras is defined as the number of 3-D points that are observed by both of them. Then we find a low-stretch spanning tree  $T_X$  in  $G_X$ . The tree  $T_X$  aims to preserve the structural information of  $G$ , and provides a reference to augment additional edges.

Second, we show how to augment additional edges to the subgraph. Suppose initially the edge set  $E_S$  is empty. For each point  $l$ , we sort  $X(l)$  according to their average distance to the other cameras in  $X(l)$  with respect to  $T_X$ . Then we pick the edges of  $E(l)$  into the subgraph according to this ordering. An edge is added into  $E_S$  if it does not induce a camera clique of size greater than  $n$ . To this end, we also maintain an array (initially set to 0, whose length is the number of cameras) which holds the size of the maximum clique that a camera belongs to. The array is updated whenever an edge is added. Repeating this process for all 3-D points results in edge set  $E_S$ .

Finally we construct the GSP- $n$  preconditioner by using all of the unary factors in the original graph and the binary factors corresponding to the edge set  $E_S$ . Note that there are two interesting special cases of the GSP- $n$  preconditioners: GSP-0 corresponds to the Jacobi preconditioner while GSP- $\infty$  corresponds to using the original graph to construct the subgraph preconditioner.

## 5.4 The Symmetry and Positive Semidefiniteness of GSP- $n$

Being symmetric and positive semi-definite (spsd) is a necessary condition for being a valid preconditioner in the conjugate gradient method. Here we show that any GSP- $n$  preconditioner is spsd. First, we know that any  $\mathbf{H} \approx \mathbf{A}^T \mathbf{A}$  matrix is always spsd, and hence GSP- $n$  is also symmetric by construction. Second, discarding off-diagonal block pairs  $A_{x_0}^T A_{l_0}$ ,  $A_{l_0}^T A_{x_0}$  in the Hessian while leaving the block-diagonal unchanged corresponds to replacing a binary factor by two unary factors in the Jacobian factor graph. The replaced binary factor corresponds to  $\mathbf{A}$ 's block-row with nonzero blocks  $A_{x_0}$  and  $A_{l_0}$ , while each new unary factor contains exactly one of these blocks. The inner product of the new factor matrix with itself is spsd, which guarantees the validity of GSP preconditioners. Note that discarding symmetrical off-diagonal entries of an *arbitrary* spsd matrix may not produce a spsd matrix. In the scalar case, Boman et al. [5] proved that matrices with this property must admit a factorization  $\mathbf{A}^T \mathbf{A}$ , with  $\mathbf{A}$  having a factor width  $\leq 2$ .

## 6 Results

### 6.1 Configurations

Here we compare the sparse factorization method (DBA) and the conjugate gradient (CG) method with three preconditioners: (1) the block Jacobi preconditioner (JACOBI), (2) the subgraph preconditioner (SPCG), and (3) the generalized subgraph preconditioner (GSP- $n$ ). The number attached to "GSP- $n$ " indicates the maximum clique size allowed in the greedy algorithm.

We use the Levenberg-Marquardt method as the nonlinear solver. The stopping criteria are (1) the number of iterations exceeds 20, (2) the average reprojection error is less than 0.8 pixel, or (3) the relative decrease of the error is less than  $10^{-2}$ .

For the linear solvers, DBA uses the *cholmod* package [9] with an approximate minimum degree ordering. For the solvers using the CG method, we solve Equation (6) by using the least-squares variant of CG [4] without forming the normal equation (see Algorithm 1). The stopping criteria for the CG method are (1) the number of iterations exceeds 2000, (2) the relative decrease of residual is less than  $10^{-2}$ .

For JACOBI, we accumulate all unary factors for each variable (i.e., the diagonal blocks of  $\mathbf{A}^T \mathbf{A}$ ) and solve them independently. For SPCG, we use the Sparse QR factorization package [10]. For GSP- $n$ , we use the *cholmod* package [9] with an ordering in which the 3-D points are eliminated first and the cameras are eliminated according to the topological ordering of the camera low-stretch spanning tree. We use Alon et al.'s algorithm to find a low-stretch spanning tree in the camera graph [3]. Note that for SPCG and GSP- $n$ , the topology of the subgraph is determined at the beginning, and never changed during the optimization.

We run the experiments on the *bal* datasets released by Agarwal et al. [2]. Since *bal* contains many datasets and some of them cannot fit into the memory of a regular PC, we select ten proper datasets from *bal* which have 100K to 500K points (see Table 2). We run all of the experiments on a Core2 Duo PC with 8G RAM.

---

**Algorithm 1.** Preconditioned Conjugate Gradient Least-Squares Method

---

**Input:** Let  $A$  be the Jacobian matrix,  $R^T R$  be the factorized preconditioner,  $x_0$  be an initial estimate,  $\epsilon$  be the tolerance, and  $t$  be the maximum number of iterations.  
 $r_0 = b - Ax_0, p_0 = s_0 = R^{-T}(A^T r_0), \gamma_0 = \|s_0\|_2^2$

**for**  $k = 0$  **to**  $t$  **do**

**if**  $\gamma_k < \epsilon$  **then break**

$t_k = R^{-1} p_k$

$q_k = A t_k$

$\alpha_k = \gamma_k / \|q_k\|_2^2$

$x_{k+1} = x_k + \alpha_k t_k$

$r_{k+1} = r_k - \alpha_k q_k$

$s_{k+1} = R^{-T}(A^T r_{k+1})$

$\gamma_{k+1} = \|s_{k+1}\|_2^2$

$\beta_k = \gamma_{k+1} / \gamma_k$

$p_{k+1} = s_{k+1} + \beta_k p_k$

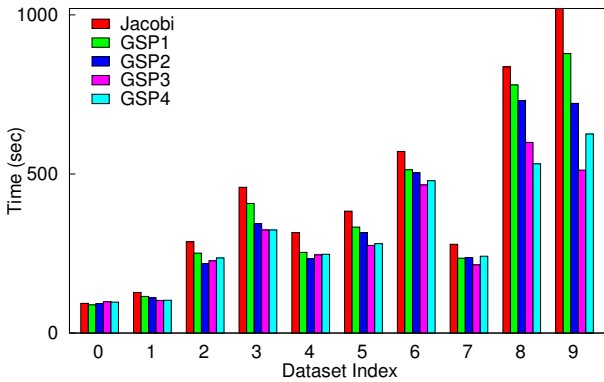
**end**

---

### 6.2 The Performance of GSP- $n$

We first investigate the performance of GSP- $n$  for different values of  $n$ , and show the timing results in Figure 12. Notice that GSP- $n$  is equivalent to JACOBI when  $n = 0$ . We exclude the linearization time and focus on comparing the linear solvers. The results show that GSP- $n$  converges faster than JACOBI by 10-30% in most cases.

We also observe that as  $n$  increases, the overall time decreases at first, but increases if  $n$  is set too high. To better understand the behavior of GSP- $n$ , we break down the timing results of one dataset and show the major components in Table 1. We can see that as  $n$  increases, the subgraph becomes denser and harder to solve, but the time spent on building the subgraph preconditioner is not significant when  $n$  is small. Here the important parts are (1) the time to apply the preconditioner per CG iteration, and (2) the number of total CG iterations. The former increases because the preconditioner



**Fig. 12.** Timing results of JACOBI and GSP- $n$  on *bal*

**Table 1.** Timing results of GSP- $n$  on the "F-05" dataset. We only show the components relevant to the linear solvers. The columns indicate (1) the maximum clique size in GSP- $n$ , (2) the percentage of edges used in the subgraph, (3) the time of building the subgraph, (4) the time per CG iteration, and (5) the number of total CG iterations, and (6) the total time.

n	edges (%)	build (s)	time/iter (s)	#iters	total (s)
0	0.0	27.2	0.48	1438	732.6
1	19.8	33.4	0.53	1130	648.8
2	26.6	48.7	0.56	866	550.5
3	32.5	69.1	0.62	631	<b>473.7</b>
4	39.0	101.5	0.78	526	512.8

**Table 2.** Timing results (secs) of the four methods on ten *bal* datasets. The second column corresponds to the name and index in the original *bal*: "D" for "Dubrovnik", "L" for "Ladybug", "V" for "Venice" and "F" for "Final".

Set	Source	Cameras	Points	Measurements	DBA	JACOBI	SPCG	GSP-3
0	V-01	89	110,973	562,976	<b>42</b>	84	401	89
1	F-01	394	100,368	534,408	<b>79</b>	113	256	96
2	V-02	245	198,739	1,091,386	<b>155</b>	245	415	196
3	D-15	356	226,730	1,255,268	<b>187</b>	397	804	285
4	V-03	427	310,384	1,699,145	313	273	695	<b>212</b>
5	L-30	1,723	156,502	678,718	578	312	718	<b>223</b>
6	V-04	744	543,562	3,058,863	886	506	913	<b>407</b>
7	F-03	961	187,103	1,692,975	1148	252	741	<b>191</b>
8	F-02	871	527,480	2,785,977	1939	776	1154	<b>564</b>
9	F-05	3,068	310,854	1,653,812	3504	894	2035	<b>473</b>

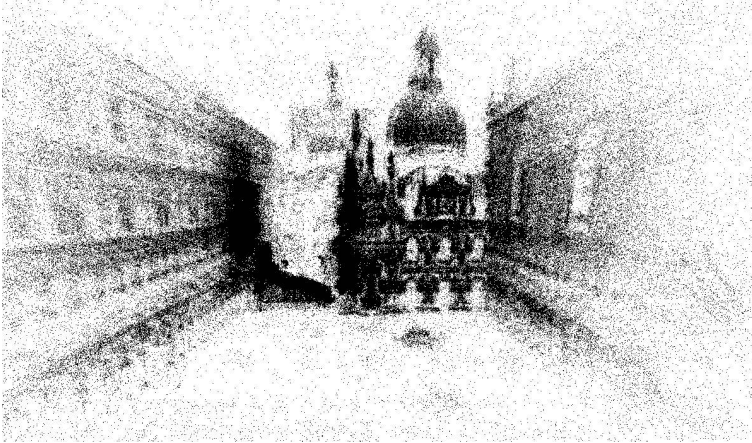
becomes denser and hence more computation is involved in the back substitution. The latter decreases because the linear systems become better conditioned. We can see that their product dominate timing and clearly there is a trade-off between these two factors.

### 6.3 Timing Results

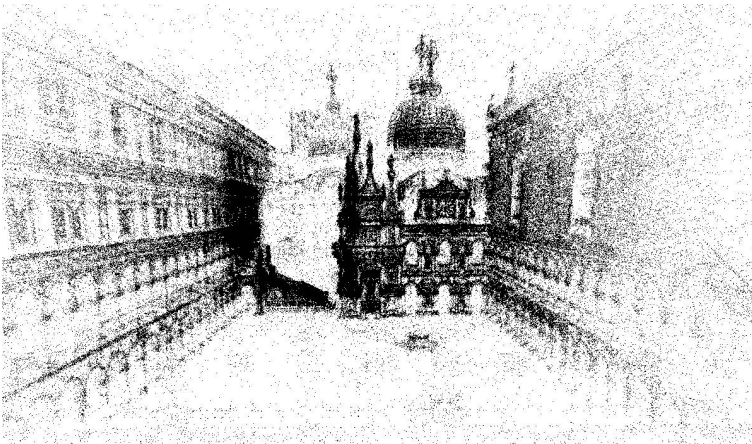
Here we compare the timing results of four linear solvers on the *bal* datasets. We use  $n = 3$  to build subgraphs for both SPCG and GSP- $n$ . The timing results in Table 2 are sorted according to the DBA time, which reflects the intrinsic difficulty of the datasets. The results confirm that sparse direct methods are efficient for small datasets, but iterative methods are better alternatives for large datasets. Comparing JACOBI and GSP, the results show that by adding extra factors to the subgraph, GSP provide better

**Table 3.** The condition numbers of the SPCG, JACOBI, and GSP-3 on three *bal* datasets

Set	Original	SPCG	JACOBI	GSP-3
D-15	5.58e+21	1.87e+06	5.94e+04	4.36e+03
V-02	6.54e+21	6.46e+09	6.35e+05	1.38e+05
F-01	3.68e+11	1.92e+08	7.54e+06	8.71e+05



(a)



(b)

**Fig. 13.** Visualization of the “F-03” datasets. The solutions obtained from solving (a) the subgraph and (b) the original graph. Similar to Figure 7 the solution to subgraph serves as a good preconditioner to solve the original problem.

preconditioners than JACOBI in most of the cases. Comparing SPCG and GSP, the results show that being able to add more unary factors to the graph is crucial to improve the convergence speed of the CG method. An example of the result is shown in Figure 13.

## 6.4 The Condition Numbers

The condition number is a common measure to estimate the convergence speed of the conjugate gradient method [21]. Here we compare the condition numbers of the linear systems preconditioned by the SPCG, JACOBI and GSP-3 preconditioners on several medium *bal* datasets. The results are shown in Table 3. We can see that the original condition numbers are huge, which indicate the slow convergence of using a plain CG solver. The SPCG preconditioner works to some extent, but is not as good as JACOBI and GSP-3. The condition numbers of GSP-3 are 5-10 times smaller than JACOBI.

## 7 Conclusions and Future Work

While direct methods are efficient for small datasets and iterative methods are more appropriate if the memory requirement is of concern, a subgraph-based preconditioning method combines their advantages and provides a better alternative for solving large-scale bundle adjustment. One such method is SPCG, which to the best of our knowledge has not been applied to the bundle adjustment problem until now. Although for large datasets SPCG is significantly better than direct methods and the plain CG method, its behavior is sub-optimal: as the bundle adjustment graph is bipartite and unbalanced, SPCG over-estimates the uncertainty of the variables. In contrast, GSP avoids this problem, and is more expressive and suitable for bundle adjustment. Well-known preconditioners like Jacobi fit naturally in the GSP context. To exploit the graphical structure of the problem, we develop an efficient algorithm rooted in combinatorial preconditioning, to construct a family of subgraph preconditioners. When applied to large datasets, the GSP- $n$  preconditioners display promising performance.

For future work, first we would like to develop a more expressive factor representation to explain and understand the other matrix preconditioners such as the Incomplete Factorization and the Symmetric and Successive Over-Relaxation preconditioners. The second is to develop a better algorithm to construct the subgraph preconditioners, and provide theoretical guarantees for their performance.

## References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: IEEE 12th International Conference on Computer Vision, pp. 72–79 (2009)
2. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle Adjustment in the Large. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 29–42. Springer, Heidelberg (2010)

3. Alon, N., Karp, R., Peleg, D., West, D.: A graph-theoretic game and its application to the k-server problem. *SIAM Journal on Computing* 24(1), 78–100 (1995)
4. Björck, A.: *Numerical Methods for Least Squares Problems*. SIAM Publications (1996)
5. Boman, E., Chen, D., Parekh, O., Toledo, S.: On factor width and symmetric h-matrices. *Linear Algebra and its Applications* 405, 239–248 (2005)
6. Boman, E., Hendrickson, B.: Support theory for preconditioning. *SIAM Journal on Matrix Analysis and Applications* 25(3), 694–717 (2003)
7. Byröd, M., Åström, K.: Bundle adjustment using conjugate gradients with multiscale preconditioning. In: *British Machine Vision Conference* (2009)
8. Byröd, M., Åström, K.: Conjugate Gradient Bundle Adjustment. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*. LNCS, vol. 6312, pp. 114–127. Springer, Heidelberg (2010)
9. Chen, Y., Davis, T., Hager, W., Rajamanickam, S.: Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software* 35(3), 1–14 (2009)
10. Davis, T.: Algorithm 915, SuiteSparseQR: multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Transactions on Mathematical Software* 38(1) (2011)
11. Dellaert, F., Kaess, M.: Square root sam: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research* 25(12), 1181–1203 (2006)
12. Dellaert, F., Carlson, J., Ila, V., Ni, K., Thorpe, C.E.: Subgraph-preconditioned conjugate gradient for large scale slam. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010)
13. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010)
14. Jeong, Y., Nister, D., Steedly, D., Szeliski, R., Kweon, I.: Pushing the envelope of modern methods for bundle adjustment. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1474–1481 (2010)
15. Jian, Y.D., Balcan, D.C., Dellaert, F.: Generalized subgraph preconditioners for large-scale bundle adjustment. In: *IEEE 13th International Conference on Computer Vision* (2011)
16. Konolige, K., Garage, W.: Sparse sparse bundle adjustment. In: *Proc. of the British Machine Vision Conference* (2010)
17. Lourakis, M., Argyros, A.: SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software* 36(1), 1–30 (2009)
18. MacKay, D.: *Information theory, inference, and learning algorithms*. Cambridge Univ. Press (2003)
19. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3D reconstruction. In: *IEEE 11th International Conference on Computer Vision* (2007)
20. Olson, E., Leonard, J., Teller, S.: Fast iterative alignment of pose graphs with poor initial estimates. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2262–2269 (2006)
21. Saad, Y.: *Iterative methods for sparse linear systems*. Society for Industrial Mathematics (2003)
22. Snavely, N., Seitz, S.M., Szeliski, R.S.: Skeletal graphs for efficient structure from motion. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2008)

23. Snavely, N., Seitz, S., Szeliski, R.: Modeling the world from internet photo collections. *International Journal of Computer Vision* 80(2), 189–210 (2008)
24. Spielman, D.A.: Algorithms, graph theory, and linear equations. In: *International Congress of Mathematicians* (2010)
25. Trefethen, L., Bau, D.: *Numerical linear algebra*, vol. 50. Society for Industrial Mathematics (1997)
26. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle Adjustment – A Modern Synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) *ICCV-WS 1999*. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)



# Achievements and Challenges in Recognizing and Reconstructing Civil Infrastructure

Ioannis Brilakis<sup>\*</sup>, Fei Dai, and Stefania-Christina Radopoulou

School of Civil and Environmental Engineering, Georgia Institute of Technology, 328 Sustainable Education Building, 790 Atlantic Dr. NW, Atlanta, GA 30332-0355, USA  
{brilakis, feidai, srado}@gatech.edu

**Abstract.** The US National Academy of Engineering recently identified restoring and improving urban infrastructure as one of the grand challenges of engineering. Part of this challenge stems from the lack of viable methods to map/label existing infrastructure. For computer vision, this challenge becomes “How can we automate the process of extracting geometric, object oriented models of infrastructure from visual data?” Object recognition and reconstruction methods have been successfully devised and/or adapted to answer this question for small or linear objects (e.g. columns). However, many infrastructure objects are large and/or planar without significant and distinctive features, such as walls, floor slabs, and bridge decks. How can we recognize and reconstruct them in a 3D model? In this paper, strategies for infrastructure object recognition and reconstruction are presented, to set the stage for posing the question above and discuss future research in featureless, large/planar object recognition and modeling.

**Keywords:** recognition, reconstruction, infrastructure, buildings, construction.

## 1 Introduction

“Restoring and Improving Urban Infrastructure” has been recently identified as one of the grand challenges of engineering in the 21st century by the National Academy of Engineering [1]. Part of this challenge stems from the lack of viable methods to map and label existing infrastructure. For computer vision, this challenge leads to the question: “How can we automate the process of extracting geometric, object oriented models of infrastructure from visual data?”

Currently, over two thirds of the effort needed to model even simple infrastructure is spent on manually converting a cloud of points to a 3D model [2, 3]. The result is that only very few constructed facilities today have a complete record of as-built information and that as-built models are not produced for the vast majority of new construction and retrofit projects, which leads to rework and design changes [1] that cost up to 10% of the installed costs [4, 5]. Any effort towards automating the modeling process will increase the percentage of modeled infrastructure projects and, considering that construction is a \$772 billion industry [6], each 1% of increase can lead up to \$772 million in savings.

---

<sup>\*</sup> Dagstuhl seminar 11261 "Outdoor and Large-Scale Real-World Scene Analysis".

From the perspective of civil engineering, this paper summarizes the achievements and challenges in recognizing and reconstructing civil infrastructure. First, it outlines current practices for as-built 3D modeling of civil infrastructure and recent research efforts in this direction. This is followed by a summary of recent research in infrastructure recognition and reconstruction, together with an outline of research solutions proposed in the authors' group ranging from the creation of Visual Pattern Recognition (VPR) models, videogrammetric progressive site modeling, to reciprocal reconstruction and recognition for modeling infrastructure. The challenge and future work in featureless, large/planar object recognition and modeling is finally discussed.

## 2 As-Built 3D Modeling of Civil Infrastructure

The current state-of-the-art approach to collecting, organizing and integrating as-built data of a constructed facility into a single data structure is to model it using building information modeling (BIM) tools [7]. This approach generates parametric building models by producing logical building objects and their parametric relationships. The as-built modeling process can be divided into three phases; during the first phase, modelers collect spatial and visual data on site through cutting-edge surveying technologies, such as laser scanning (LIDAR) and photo/videogrammetry. The resulting data is in the form of images and a high-resolution point cloud that contains the spatial information of all elements in the scene. During the second phase, the 3D point cloud is replaced with objects and object relationships. This is achieved by having an operator observing the data manually to a) identify each object type, b) search for it in a database of standardized objects, c) fit it in the point cloud with partial help from fitting algorithms for optimal fitting, and d) assign the relationships of each object with the rest. The third phase, which is also manual, includes the assignment of any non-spatial as-built attributes (e.g., material, schedule, costs) to each object. The key difference of the outcome of the first phase (point cloud) with the final result (object oriented model) is that the final model contains multiple, discrete elements with a wide range of attributes (i.e. material, schedule, cost and other information). This is why the last two phases are necessary to derive the full benefits of the resulting model.

Although as-built modeling is significantly assisted by recent technological advancements, most of it remains manual (Figure 1) making it time-consuming and costly. Professional modelers such as VECO [24] and Reality Measurements [25] report similar findings. This problem cascades further, since the significant cost and effort needed to convert the sensed infrastructure points to the desired object model counteracts the benefits of spatial modeling for the majority of civil infrastructure. This is true especially for small civil projects, where the projected savings hardly justify adopting this technology [26]. As a result, the infiltration of innovative spatial modeling technologies to the Architecture, Engineering and Construction (AEC) industry is slower than expected [27].

Automation primarily enables the recovery of the 3D points [8, 9, 10, 11, 12, 13, 14, 15]. Recent research efforts have attempted to automate some of the manual steps by matching certain types of as-designed CAD objects with the point cloud [16, 17, 18, 19]. Others have tried to classify and label components (i.e., wall, floor, window,

ceiling, floor plan, and molding) of a building room on a point cloud assuming that the contextual relationships of the indoor objects are orthogonal, parallel, adjacent or coplanar [20, 21, 22, 23]. Further automation could be achieved with the help of object recognition and reconstruction. A review of these techniques developed in the computer vision and civil infrastructure fields follows.

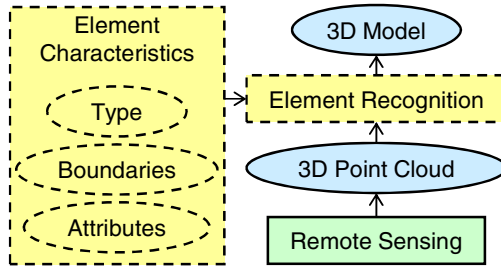


Fig. 1. Dashed lines indicate manual steps

### 3 Recognition of Civil Infrastructure

Recognizing infrastructure objects can be generally divided into two categories based on the type of input data utilized: spatial or visual data.

Spatial data in this case is typically in the form of high-resolution 3D point clouds, collected by remote sensing technologies such as laser scanning, 3D range cameras, or photo/videogrammetry. Methods utilizing spatial data [28, 29, 21] explore a priori knowledge with respect to object semantics (e.g., it is known that a point cloud contains walls, floors, ceilings, etc.), geometrical constraints (e.g., saddleback roofs have an angle between two planes), and spatial relationships (e.g., a wall is orthogonal to and connects with floor and ceiling) of objects to distinguish between different types of objects. The advantage of such methods is that even objects that are geometrically similar (e.g. walls, ceilings), and otherwise difficult to differentiate if observed in isolation, can be differentiated [22]. Those methods have been actively explored in building objects detection [30, 20, 21, 23] based on the observation that the majority of building objects can be decomposed into parts that correspond to geometric primitives such as planes, cylinders, and spheres, and have strong spatial relationships such as orthogonality, parallelity, and coplanarity. In the work of Huber et al. [21], for example, the point cloud of a building's interior is first projected horizontally to identify ceilings and floors based on the observation that the projected point density is highest at those locations; once the planes of the ceiling and floor are identified, the point cloud is then projected onto the floor to identify the walls according to the same criterion. However, one major limitation of these methods is each type of object needs unique encoding [30], leading so far to the detection of limited objects (i.e., wall, floor, window, ceiling, floor plan, and molding) in restricted scenarios. Also, these methods have not yet been tested in highly cluttered scenes (e.g., construction site).

Alternative methods start from visual data. Visual data in this case is typically in the form of images and videos. Object detection utilizing visual data in most cases is model based. A priori knowledge of a numerical model containing several distinctive visual features of an element type is needed for the machine to recognize other elements of the same type. These object models consist of a collection of several spatially correlated characteristics of an element. So far research efforts have led to a number of numerical based models such as constellation models [78, 79, 80], pictorial structures [81, 82], shared feature models [83], and pyramid structures [84]. These models have emphasized on modeling the shape and appearance variability of objects viewed from specific poses [78, 85] or a mixture of poses [83, 86, 87, 88], and describe the objects by a list of class labels, such as chairs, desks etc., together with their rough 2D location and scale. However, these models are not capable of estimating the 3D position of objects in relation to each other and the observer, and are limited to be operational within specific view point configurations, such as front, back and  $\frac{3}{4}$  views. Methods that are able to detect single objects from different poses exist [89, 90, 91, 92, 93, 94], but they are unable to detect object categories. Lately, there is a new class of methods that seek to detect objects from true multi-view settings [75, 76, 77, 95, 96, 97, 98, 99, 100, 101, 102, 103]. In these methods, object elements (features, parts, contours) are connected across views to form a unique, coherent model for the object category.

For civil infrastructure, research in object recognition and classification using visual data is also a topic of significant interest in recent years. For example, Shin and Hryciw [31] determined average grain size from soil mass images using a two-dimensional wavelet decomposition method. Masad et al. [32] and Pan and Tutumluer [33] created a 3D image analyzer to determine coarse aggregate size, texture and angularity. Lee et al [34] created an automated, image-based steel bridge corrosion detection method. Jeong and Abraham [35] evaluated underground imaging techniques for underground infrastructure detection. Hutchinson and Chen [36] created an automated statistical-based procedure for image based concrete damage evaluation. Lester and Bernold [37] used translation invariant wavelet packet detection to filter ground penetration radar data for characterizing buried utilities. Chen et al. [38] created an adaptive ellipse approach for the automated detection of bridge coating in images. Chae et al [39], Costello et al. [40], Sinha and Fieguth [41], Yang and Su [42], and Guo et al. [43] created automated pipe condition assessment methods using imaging techniques. Golparvar-Fard et al. [13] matched certain types of as-designed CAD objects with time-lapse photographs for progress monitoring and visualization. Zhu and Brilakis [44, 45] created two defect detection methods for concrete inspection. Son and Kim [19] created a method that is capable of recognizing simple 3D structural components (i.e. beams and columns) with specific configurations. These, and many other highly successful efforts, reflect the great importance of applying vision technologies to solve civil infrastructure related identification and assessment problems. Nonetheless, these undertakings are still at the basic stage that most efforts are primarily focused on recognizing attributes (e.g., soil mass, cracks, air pockets, discoloration), and materials (e.g., concrete, steel, wood). A higher level of detection methods is desired for detecting, recognizing, and classifying numerous, complex civil infrastructure objects in natural and cluttered scenes.

## 4 Reconstruction of Civil Infrastructure

3D reconstruction is the process of capturing geometry and structure (i.e., 3D coordinates) of an object in the form of 3D point clouds. It is typically used in civil engineering applications with respect to quality control and assurance, as-built quantity takeoffs, as-designed and as-built comparisons, productivity measuring, project monitoring and control systems.

Conventionally, time-of-flight laser scanners and 3D range cameras are used to collect as-built spatial data of infrastructure. Laser scanners perform fast and produce accurate results [47, 48]. However, the price of a laser scanner is usually high and it counteracts the benefit of generating a high-resolution map of depths, making this device infeasible for small construction projects, where the projected savings hardly justify purchasing this device. Regarding 3D range cameras, even though they show efficiency [49, 50], their applications suffer from short-range data collection. The measurement range of this technology is usually less than 10m [26, 51], making it infeasible for large-scale civil infrastructure projects.

In contrast to time-of-flight technologies, vision-based methods are based on spatial computation conducted in images or videos to derive 3D information of an object. This makes these methods inexpensive and easy-to-use. In the field of computer science, there is a large number of vision-based techniques that have been developed such as scale-space theory [120], epipolar geometry [8], Scale Invariant Feature Transforms (SIFT) [104], combined corner and edge detectors [105], point feature detector and tracker [106], Speeded Up Robust Features (SURF) [107], scale and affine invariant point detector [108], and Patch-based Multi-View Stereo (PMVS) [53]. Inspired by these advances, many methods that use multiple views of a scene have been developed for 3D reconstruction [8, 9, 10, 11, 12, 13, 14, 15, 89, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119]. For instance, Nistér [9] automated passive recovery of 3D scenes from images and video. Brown and Lowe [89] developed a method for unsupervised 3D object recognition and reconstruction. Snavely et al. [10] created a method for internet photo collection based modeling. Agarwal et al. [109] exploited a massive image dataset in order to recover large scale scenes. Bok et al. [110] combined vision-based and laser-scanning sensors to reconstruct historical monuments.

Typically, based on the input data utilized, the vision-based methods can be divided into two categories: photogrammetry and videogrammetry. Both have been extensively investigated and developed for use in the civil engineering field.

Regarding photogrammetric methods, Golparvar-Fard et al. [54, 55] created a four-dimensional augmented reality (D4AR) system to monitor the progress of construction projects. In their method, a collection of uncalibrated daily photos captured from construction sites are utilized to generate 3D point clouds, and the points are then superimposed into as-planned models to derive deviations of the project progress. Similarly, Ibrahim et al. [56] proposed a method to assess the process of site activities using site photos and a database of building component models. Quiñones-Rozo et al. [57] established a method to retrieve a 3D model and track the activity progress for an excavation site using site images. To continue, González-Aguilera and Gómez-Lahoz [58] created a vision-based measurement method to analyze geometric dimensions of bridges. In their method, one single

image is utilized to measure the structure by incorporating the contextual information (i.e., perpendicularity, co-planarity, and parallelism) and image invariants (i.e., distance and angles) in order to fix the absolute value of the dimension. Dai and Lu [59] evaluated the accuracy of applying photogrammetry for measuring geometric dimensions of building objects. A single off-the-shelf camera is used in their method, and a process of manually defining the length of a reference line is needed to fix the scale of the Euclidean reconstruction.

Several studies have also been conducted to develop videogrammetric methods for reconstructing 3D infrastructure. Pollefeys et al. [11] developed a 3D reconstruction system for reconstructing urban scenes from video streams. This method uses GPS (Global Positioning System) to geo-tag locations of the cameras, making it possible to model large-scale environments (e.g., cities). However, the GPS signal is weak in urban dense areas. Chae and Kano [60] created a stereo videogrammetric system to control the project progress. In their method, the data was extracted from two sequences of video streams, and commercial software was used to compute the locations of site objects. Nonetheless, their work reported high geometric errors for estimating the object's positions. Son and Kim [19] applied videogrammetry to acquire 3D data for recognizing simple structural components. In their method, data acquisition was conducted by use of a trinocular camera system.

In essence, photo/videogrammetric methods provide great potential to conventional time-of-flight-based methods in reconstructing spatial data of infrastructure. Rather than producing photorealistic visuals, however, a need for more robust and accurate reconstruction is desired for reliable use in civil engineering applications. Attention should also be given to the formalization of factors (type of camera, image resolution, shooting distance) for achieving a specific level of accuracy while maintaining run-time efficiency. Moreover, confidence measure is needed for guiding sufficient data collection. Its goal is to avoid occlusion and low quality of frames that otherwise will undermine reliable post-processing of 3D points. This is true particularly in a built environment where there is only one chance to videotape an ongoing product in its as-built state.

## 5 Proposed Solutions

This section presents recognition and reconstruction research solutions from the authors' group aimed at reverse engineering the civil infrastructure. They are the first steps in the larger agenda of automatic modeling of as-built infrastructure objects.

### 5.1 Visual Pattern Recognition Models

A process for the manual creation of Visual Pattern Recognition (VPR) models is proposed as a simple and robust way to perform model based recognition on infrastructure elements. The ultimate goal is to gradually build an infrastructure element VPR model repository.

Figure 2 depicts the proposed VPR model generation process that consists of three steps. In the first step (*identify elements of visual characteristics*), the distinctive visual characteristics that refer to specific image signal patterns are collected either

directly, through image intensity and color channels, or indirectly, through different image transformations, such as Fourier and Wavelet transforms. Such characteristics are usually related to color, texture and geometric properties, out of which the most suitable should be selected. The choice depends on how well a characteristic is modeled with the available image analysis tools and how “special” it makes the object in the scene. The total number of characteristics required for the recognition of an object varies according to the recognition performance. Only those characteristics that contribute to the increase of recognition accuracy are considered. The most distinctive characteristics for the recognition of infrastructure objects can be categorized to shape-related or texture-related types. The second step (*represent image analysis features*) involves finding or making the most suitable tool for recognizing each feature. Given that there are many ways of representing texture, such as with the use of spot filters, wavelet coefficients etc., it is necessary to perform a comparison of the accuracy of each representation of features and choose the most appropriate methods. Last, it is very important to correlate features and represent their relative topology to improve the object recognition performance. Thus, in the third step (*correlate features with their topology*), features and relative topology are bundled as an object, creating its VPR model. Each VPR model can be stored to gradually build the VPR repository.

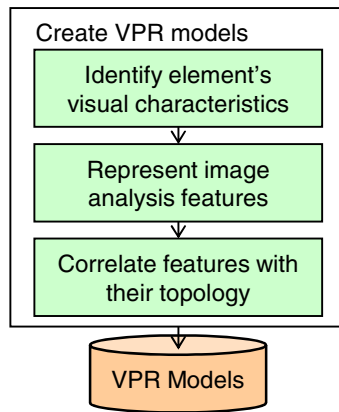


Fig. 2. VPR model creation

## 5.2 Videogrammetric Progressive Site Modeling

A videogrammetric framework for generating high-resolution (dense) point clouds is proposed as a simple and robust way to perform 3D Euclidean reconstruction on infrastructure objects. The final goal is to provide reliable and accurate 3D spatial data for automatic modeling of as-built infrastructure objects.

Figure 3 depicts the proposed framework under which as-built infrastructure objects are progressively reconstructed. At the beginning, a physical scene is sensed using a calibrated set of high-resolution video cameras, which is progressively traversed around the scene. Feature points between the left and right video frames of the stereo rig are matched. The 2D location of the matched features and camera

calibration provide enough information to calculate the 3D coordinates of the feature points, which are going to be fed to the camera pose estimation step. Matched features are then identified using the KLT tracker [61] in consecutive video frames. Since the triangulation is much more uncertain in the depth direction, the system employs Perspective-n-Point (PnP) algorithms for camera pose estimation. PnP algorithms estimate the pose of the camera set from 3D to 2D point correspondences (i.e., the 3D points reconstructed from one stereo pair and their corresponding 2D points on the image plane after camera movement). Statistical methods (e.g., Kalman filter) and global optimization techniques (e.g., bundle adjustment) are further employed, for a certain number of frames, to refine the results. Since the outcome of the previous steps is a sparse 3D point cloud, dense multi-view stereo matching algorithms (such as those proposed in [53, 62]) are required to generate dense point clouds. The proposed technique, initially, rectifies the left and right video frames for this purpose. Then, a novel adoptive window-matching algorithm is used to automatically match non-feature points. This way, the point cloud will be progressively expanded, by the addition of the 3D points in new frames, to cover the whole scene.

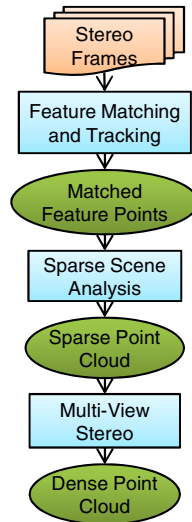


Fig. 3. Videogrammetric progressive site modeling method

### 5.3 Reciprocal Reconstruction and Recognition

In order to advance the level of detail for modeling as-built infrastructure objects, a reciprocal process that combines the ability to recognize objects from images with that of reconstructing the 3D scene is proposed. The expected outcome is the sensed spatial data of infrastructure objects associated with their structural detected members.

Figure 4 depicts the proposed process that combines reconstruction and recognition for modeling as-built infrastructure objects. According to Figure 4, a calibrated



high-resolution camera is used to videotape an infrastructure scene from all accessible angles with minimum occlusion. Based on the video frames captured, the point feature detection, matching techniques, Structure from Motion (SfM) and Multi-View Stereo (MVS) algorithms are utilized to estimate the camera trajectories and generate the dense point cloud of the scene. In parallel to this, the structural members (concrete columns and beams in this study) are detected in the resulting stream of images, and their occupying regions are marked. By roughly registering the detected regions onto the 3D point cloud using the obtained camera trajectories, the result is a rendered 3D view of the structure with the recognized 3D element boundaries marked. This loops back to the detection of structural members, which can now be performed on the spatial data covered by the visually marked 3D element boundaries, resulting in more robust and accurate element detection, and consequently improved element matching and reconstruction. The final model is expected to be an accurate 3D representation of the structure with the load bearing linear members detected marked with 3D bounding boxes. This model is provided to the modeler, who can then use it to complete the model making process.

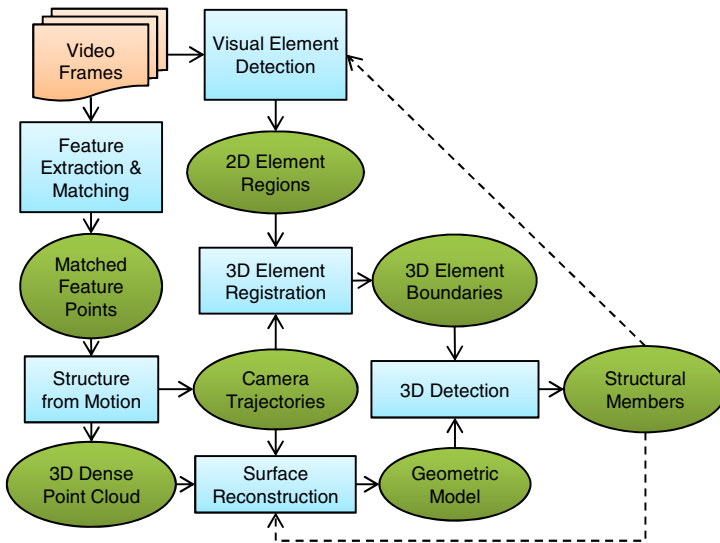


Fig. 4. Reciprocal reconstruction and recognition method

## 6 Implementation and Results

This section presents the implementation and results of each solution. All method prototypes are implemented on top of Gygax, a barebone research platform developed by the authors' research group using Microsoft C# with Windows Presentation Foundation and publicly available libraries such as EmguCV, a cross platform .Net wrapper to OpenCV for access to computer vision tools.

## 6.1 VPR Model Examples and Results

The VPR models of concrete columns, concrete cracks, air pockets, exposed reinforcement, and asphalt pavement potholes have been created so far using the proposed process depicted in Section 5.1. The detailed steps of creating these VPR models and their validations are presented below.

The evaluation of each application was performed by testing a database of images/videos for each type of civil infrastructure elements. All images were captured at natural light conditions by using digital cameras. In the case of concrete column detection, images at low-light conditions, where elements were difficult to be identified even by human naked eye, were rejected. The comparison was made between the outcomes of each VPR model and manual recognition results. In each case, the recognition precision and recall were measured. Precision measures the detection exactness and is equal to the percentage of the number of elements correctly recognized within the total number of elements correctly and incorrectly recognized. Recall measures the detection completeness and is equal to the percentage of the number of elements correctly recognized within the total number of elements correctly recognized and not recognized at all. Table 1 shows the outcomes of each application. The high recognition performance validates the effectiveness of using the method for creating VPR models to facilitate the recognition of infrastructure related elements.

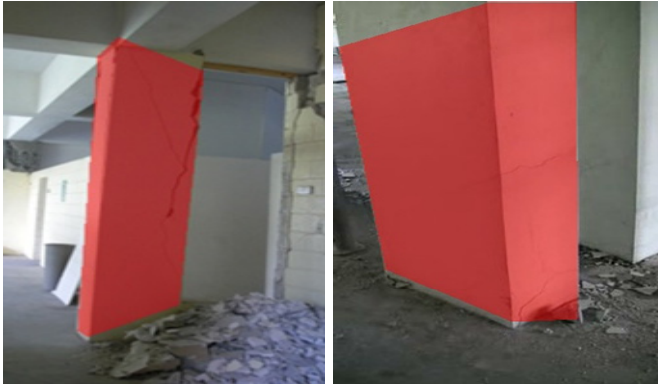
**Table 1.** Precision and recall summarization of VPR model applications

	Precision	Recall
Concrete columns	84.4%	74.5%
Concrete cracks	64.2%	91.8%
Air pockets	91.1%	85.6%
Exposed reinforcement	83.2%	82.2%
Asphalt pavement potholes	81.6%	86.1%

**Concrete Columns.** Two are the distinctive characteristics of a concrete column in an image. The first one is that each column has a pair of long near-vertical lines and the second is that the uniform texture and color patterns lie on each of the member's surface. Because of these characteristics, edge detection, Hough transform, image segmentation and a machine learning classifier are used for concrete column detection.

First, the Canny edge detector is used for producing a binary image that is composed of edge and non-edge points. The Hough transform is then applied to group the distribution of edge points and retrieve long vertical line information from the edge map. Each vertical line is compared to its neighboring ones and if two vertical lines are similar in size then they are supposed to be a pair. The comparison keeps on being performed until no pairs can be formed. The criterion of keeping a pair of lines is if their aspect ratio (width/length) is greater than one. The procedure continues by calculating the color and texture feature of the image region contained in each pair of

lines. An artificial neural network performs the material type classification. If the material belongs to concrete, then a concrete column is detected. An example of recognizing different concrete columns using this method can be seen in Figure 5. The final results show a precision and recall of 84.4% and 74.5% respectively. Further details regarding the procedure and its results can be found in [63].

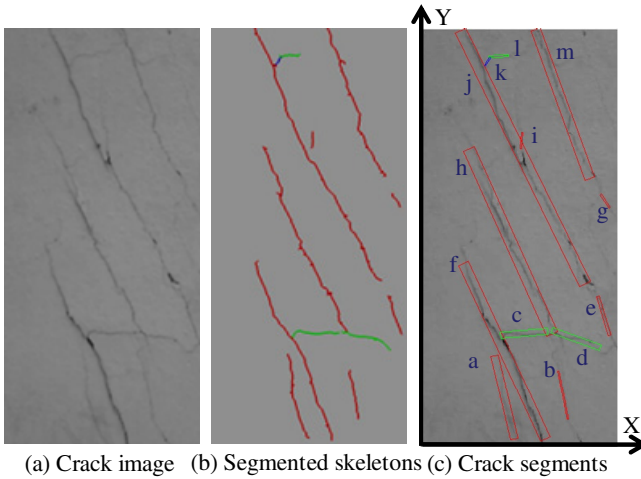


**Fig. 5.** Concrete columns detection

**Concrete Cracks.** This method aims to retrieve the properties of the cracks on concrete structural elements. The proposed framework is divided into two stages. In the first stage, which is crack detection, a crack map for every structural element surface is produced. The method used for crack detection is a modified version of the solution proposed by Yamaguchi and Hashimoto [64]. The gradient of each image pixel is initially calculated and the pixels with high gradient magnitudes are percolated.

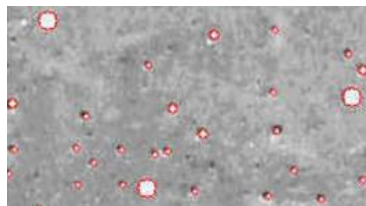
The second stage of the framework is to retrieve crack properties. Those are crack length, orientation, maximum and average width. A thinning algorithm is applied to the crack map in order to retrieve the crack's skeleton, from which crack properties are acquired. Each crack pixel is then paired with the nearest distance to its boundaries. The distance is calculated with a Euclidean distance transform. This is the information needed for retrieving the necessary properties and reconstructing the crack. An example of recognizing concrete cracks using this method can be seen in Figure 6. The final results show a precision and recall of 64.2% and 91.8% respectively. More details regarding this method can be found in [65].

**Air Pockets.** This method aims to detect air pockets on the concrete surface of infrastructure elements captured in images. The unique characteristics used in this case are the circular shape of an air pocket and its darker region in comparison to the surrounding ones.



**Fig. 6.** Crack detection

The procedure of air pocket detection is as follows. First, a spot filter is created to detect the air pockets. Three concentric, symmetric Gaussian filters with weights 1, -2 and 1 and sigmas 0.62, 1 and 1.6 form the filter. The results of the filter are expected to be high for the regions where air pockets exist and have the same size as the filter. In any other case, the values of the intensities are low. In order to detect air pockets of larger size than the filter, the image is scaled down and filtered again. Now, the air pockets previously identified can't be detected, but larger ones can. An example of recognizing different sizes of air pockets at concrete surfaces using this method can be seen in Figure 7. The final results show a precision and recall of 91.1% and 85.6% respectively. More details regarding this method can be found in [66].

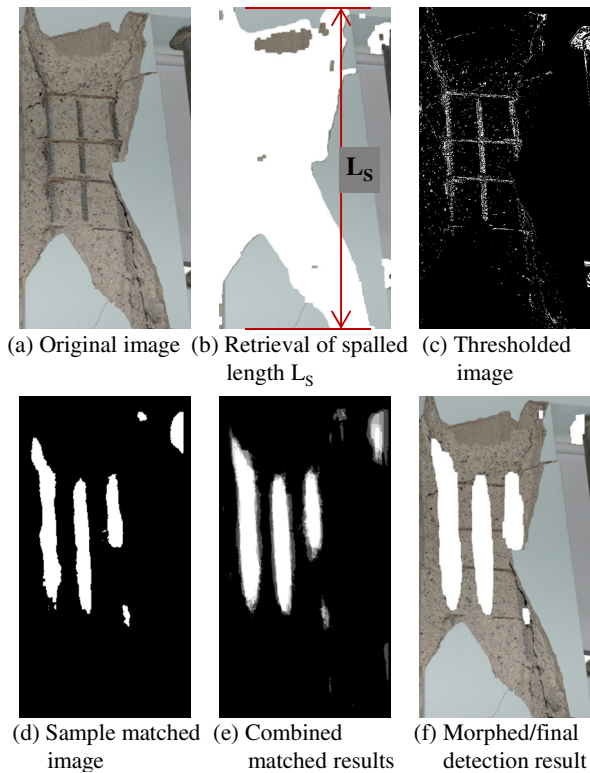


**Fig. 7.** Detection of air pockets

**Exposed Reinforcement.** The feature characteristics used for exposed reinforcement detection are: 1) such a region is darker than its surrounding ones, 2) ribbed texture exists along the reinforcement surface, and 3) the width of exposed reinforcement steel is significantly smaller than that of the concrete element.

The procedure starts by applying a threshold to the original image to get a binary version of the reinforcing area. Template matching is then strengthening the image areas of reinforcing bars. In order to make the response of the reinforcement invariant

to the orientation of the ribbed regions of the reinforcement bars, multiple templates are used. A threshold is applied again to each resulting image in order to isolate the height-intensity pixels that represent the bars. Last, vertical and horizontal profiling is applied to reject any superfluous rebar pixels. Examples of exposed reinforcement recognition can be seen in Figure 8. The final results show a precision and recall of 83.2% and 82.2% respectively. More details regarding this method can be found in [67].



**Fig. 8.** Exposed reinforcement detection

**Asphalt Pavement Potholes.** This framework aims to detect, recognize, spatially locate and evaluate the magnitude of defects of asphalt pavements. A high-speed fish-eye camera that can tilt downwards is placed on the rear of a vehicle. While the vehicle is moving forward, the camera is capturing the pavement. A computer placed in the rear of a vehicle processes the data in two stages. First, high speed (real time) algorithms are used to detect frames that might contain evidence of defects and then the selected frames are run by defect detection algorithms. The final result is frames with regions characterized by the type of defect identified. Finally, defect property measurement algorithms assess the severity of the recognized defect.

The distinctive visual characteristics identified for potholes are: 1) a pothole includes one or more shadows that are darker than the surrounding area, 2) the shape of a pothole is almost elliptical, and 3) the surface texture inside a pothole is coarser and grainier than that of the surrounding surface texture.

The proposed method is separated into three steps. First, the images are segmented. Actually, the original color images are transformed into gray-scale images and noise is reduced using a  $5 \times 5$  median filter. In order to take advantage of the first distinctive visual characteristic, a histogram shape-based thresholding algorithm is used.

Then, shape extraction is performed. Regions that either have a linear shape or are connected to the boundary of the image are rejected since they are assumed not to be potholes. In order to do the above, the length of the major axis ( $l_{max}$ ), the position of the centroid ( $P_{cent}$ ) and the orientation angle ( $\alpha$ ) are determined. Having this information, regions are separated into those that might represent a pothole shade and those that could represent an entire pothole. A sequence of morphological processes is performed (thinning and skeleton branching) to approximate the elliptical shape of a pothole.

Finally, the texture of all regions is described using the standard deviation of gray-level intensity values as a statistical measure. Then each region texture is compared with that of the surrounding region to identify false candidates and true potholes. During this step, three spot filters of the Leung-Malik (LM) filter bank [68] and one spot filter of the Schmid (S) [69] filter bank are applied to the images. Examples of asphalt pavement pothole recognition can be seen in Figure 9. The final results show a precision and recall of 81.6% and 86.1% respectively. More details regarding this method can be found in [70].



**Fig. 9.** Detection of asphalt pavement potholes

## 6.2 Generating 3D Point Clouds of Site Objects

The equipment used to assemble the prototype for this implementation were a set of multi-mega pixel resolution Canon Vixia HF S100 cameras, which were calibrated with the use of Bouguet's stereo camera calibration toolbox [71]. The intrinsic and extrinsic parameters of the camera set were estimated by a set of stereo images of a chessboard that is consisted of 30×30 mm squares. After the camera calibration, part of the application was tested in a control, yet realistic setting, but most of it was tested in a real built environment. The resolution of the cameras was set to 1600×1200 pixels in both environments.

SURF features were extracted from individual frames in a dataset of stereo pairs of video frames. The feature matching was performed by implementing the Euclidean distance between the descriptor vectors. The average ratio of correct matches was 97.74%. The RANSAC algorithm was used for discarding incorrect matches. The fundamental matrix was used as the mathematical model and the probability of selecting inliers from the dataset was set to 99%.

A sparse 3D point cloud was then produced for each pair of stereo frames. The 3D coordinates of the feature points were found by visual triangulation. In order to test the validity of the generated point cloud, the spatial distance between randomly selected points in the point cloud and their corresponding tape measurements in the real world were compared. The average value was 4.7 mm and the standard deviation 24.9 mm.

A dense matching map between points of different views is generated with an adaptive window-matching algorithm. Figure 10 shows the result of this method. The total processing time of a regular netbook for each frame set was 1.2 min. Further details regarding the procedure and its results can be found in [72].

## 6.3 Reciprocal Recognition and Reconstruction for Bridge Modeling

A monocular videogrammetric pipeline has been created in order to achieve dense 3D reconstruction of concrete bridges. In order to improve the accuracy of the resulting 3D points, three camera motion estimation algorithms were compared by using a consumer-grade camera (i.e., 8 megapixel Nikon Coolpix L19 in this study). Given that the baseline was set as 60 cm and the depth as 12 m, the average translation and rotation errors of camera ego-motions for three algorithms were measured. Figure 11 shows the experimental results. As shown in Figure 11, the 5-point algorithm resulted in the most accurate outputs. Further details regarding the comparison procedure and its results can be found in [73]. Moreover, a monocular videogrammetric prototype has been implemented. It is robust even when motion blur exists in the video frames, and is capable of obtaining an optimum selection from a lengthy video stream, sufficing to generate a dense 3D point cloud and saving computational resources (e.g., CPU, memory, processing time) needed for post-processing of the video frames. Figure 12 gives a snapshot of the dense point cloud of a concrete bridge.



Fig. 10. 3D reconstruction of site construction application

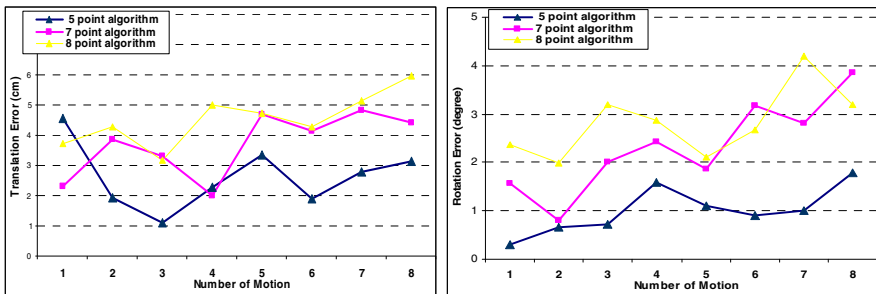
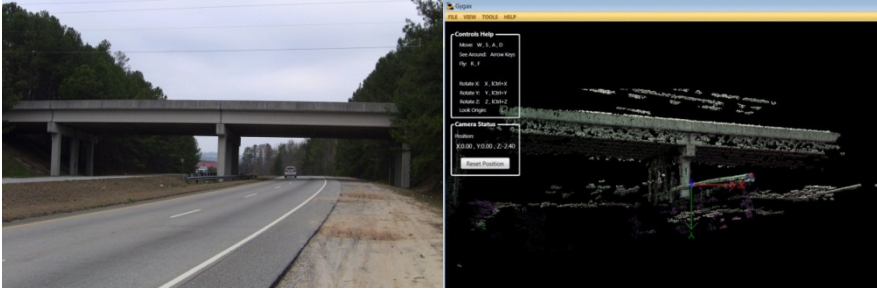


Fig. 11. Evaluation of camera motion estimation algorithms





**Fig. 12.** Snapshot of a produced dense point cloud (right) of a concrete bridge (left)

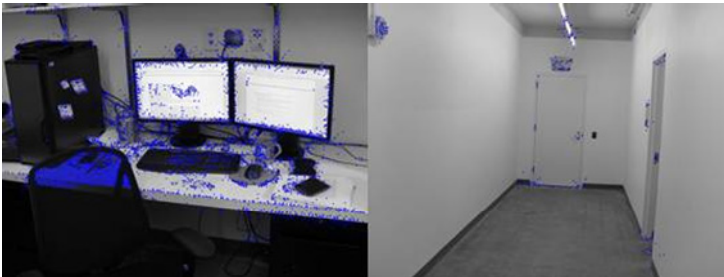
## 7 Challenges and Future Directions

While visual features are useful for identifying the type of an object and inferring its properties such as texture, shape, structure, and geometry, research in object detection using visual features in the area of civil infrastructure is still at the basic stage of recognizing attributes (e.g., soil mass, cracks, air pockets, discoloration), materials (e.g., concrete, steel, wood), and pose-invariant object types (e.g. columns) of objects without determining the 3D arrangement of the scene. At the same time, detecting simple 3D structural components (i.e. wall, floor, window, ceiling, and molding) of a building interior on a point cloud has been attempted [30, 20, 21, 22, 23] and may have significant potential. However, these methods have so far only been applied and tested in restricted scenarios (i.e. an interior of a room) requiring a priori knowledge (e.g., walls are adjacent and orthogonal to floors) without considering occlusions (furniture) and utilization of visual features (color, texture, etc.) of the objects, leading to their constrained applicability in complex civil infrastructure scenes. Also, the VPR models are not view or scale-invariant [74]. Consequently, they work very well for view/scale invariant objects (columns) or planar patterns (damage, defects) from given distance ranges, but are not effective for any other type of objects without having to make a separate model for each view and scale. In any case, these efforts are all great initial steps towards model based civil infrastructure elements detection. Considering the sheer volume of infrastructure elements that exist and complexity of their types, a more general numerical representation strategy of the appearance of infrastructure elements is needed. Even beyond that, if such representations were made available in the future, there are currently no robust methods for matching the elements of these representations with a 3D point cloud.

In pursuing robust and accurate infrastructure object recognition and reconstruction, expanding and customizing existing efforts is highly necessary. Current reconstruction methods primarily rely on applying scale invariant feature detectors, such as SIFT to match across images. These detectors find points or patches that are robust to image translation, scaling, and rotation. This is reasonable for generic, natural scene reconstruction. However for civil infrastructure scenes, most building elements lack such distinctive affine invariant features [12, 51], which makes

feature detection methods ineffective, and therefore the reconstruction process incapable of tracking camera trajectories and reconstructing 3D structure of an object. Similarly, the lack of sufficient visual features also undermines the performance of current prevailing generic multi-view models [75, 76, 77] in recognizing infrastructure-related objects. Generic multi-view models are developed in the computer vision area and are capable of recognizing objects under arbitrary viewing conditions as well as recovering the basic geometrical attributes of object categories relative to the observer and the environment. Typically, such models employ learning processes that need a set of object parts that represent visually distinctive characteristics in images for training, and it is achieved by applying the aforementioned affine invariant feature detectors. The result is that many infrastructure objects that are large and/or planar without significant and distinctive features are excluded from the use of the generic multi-view models. Figure 13 shows a typical scene of a monitor, keyboard, mouse, cup, etc. that can have up to 5159 features while the building corridor scene (right) only contains 256 features, most of which are not located on the building objects, i.e., wall, floor, ceiling, and door.

Moreover, practical challenges lie in advancing current reconstruction algorithms to produce more than nice visuals. Civil engineering applications also have the need of formalizing factors that achieve specific levels of accuracy and run-time efficiency simultaneously. In addition, confidence measures for guiding sufficient data collection is required to avoid occlusion and low quality frames for reliable post-processing of 3D points. This is true particularly in a built environment where there is only one chance to videotape an ongoing product in its as-built state.



**Fig. 13.** Detection in generic (left) & infrastructure (right) scenes

## 8 Conclusions

As-built spatial modeling is the process of capturing the infrastructure's spatial data and transforming it into a structured, object-oriented representation suitable for generating useful information for solving complex problems. Nowadays, the greatest part of as-built modeling procedures is manual, thus inefficient. Researchers both in the fields of civil infrastructure and computer vision are geared toward finding automated solutions.

This paper reviewed and summarized recent research efforts in recognition and reconstruction in civil engineering applications. This was followed by a presentation of different solutions proposed in the authors' group. First, a VPR framework, along with the gradual build of a VPR repository was proposed. This framework is applied for the modeling of concrete columns, concrete cracks, air pockets, exposed reinforcement and asphalt pavement potholes. Then a framework for videogrammetric progressive site modeling is presented. This was tested in the 3D Euclidean reconstruction of a site construction. Last, a method for reciprocal reconstruction and recognition is proposed. A road bridge was selected for validating this method, as a first step to demonstrate the effectiveness of producing a 3D geometric model with a single camera.

Current methods in recognizing and reconstructing civil infrastructure-related objects are still at the basic stage. While it is true that computer vision concepts have significant potential in this, there is still much to be desired for attaining a robust and accurate recognition and reconstruction of civil infrastructure, particularly for those that are large and/or planar without significant and distinctive features. Yet, there is plenty of knowledge that can be harvested as computer vision technologies have greatly matured, and tasks that were considered to be impossible ten years ago are now within reach.

**Acknowledgements.** This presented research was funded by the U.S. National Science Foundation (NSF) under Grants #0948415, 0800170, 0943112 and 1000700. The authors gratefully acknowledge NSF's support. The authors would also like to acknowledge Zhenhua Zhu, Christian Koch, Abbas Rashidi, Habib Fathi, Gauri Jog, and Stephanie German for their contributions to the research presented above. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF, and the individuals named above.

## References

1. National Academy of Engineering, <http://www.engineeringchallenges.org/Object.File/Master/11/574/Grand%20Challenges%20final%20book.pdf>
2. Jaselskis, E.J., Gao, Z., Walters, R.C.: Improving transportation projects using laser scanning. *Journal of Construction Engineering & Management* 131(3), 377–384 (2005)
3. Brilakis, I., German, S., Zhu, Z.: Visual Pattern Recognition Models for Remote Sensing of Civil Infrastructure. *Journal of Computing in Civil Engineering* 25(5) (2011) (in press)
4. Reddington, J.: Leica Geosystems HDS Plant Seminar, [http://www.wipco.co.kr/2005\\_Data/Korea%20plant%20oct05.pdf](http://www.wipco.co.kr/2005_Data/Korea%20plant%20oct05.pdf)
5. Sanders, F.H.: 3D Laser Scanning Helps Chevron Revamp Platform. *Oil & Gas Journal* 99(18), 92–98 (2001)
6. Census Bureau, <http://www.census.gov/const/C30/release.pdf>
7. Eastman, C., Teicholz, P., Sacks, R., Liston, K.: *BIM Handbook: A Guide to Building Information Modeling*. Wiley, New Jersey (2008)
8. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2004)

9. Nistér, D.: Automatic Passive Recovery of 3D from Images and Video. In: 2nd International Symposium on 3D Data Processing, Visualization & Transmission, pp. 438–445. IEEE Press, Washington (2004)
10. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the World from Internet Photo Collections. *International Journal of Computer Vision* 80(2), 189–210 (2008)
11. Pollefeys, M., Nister, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, R., Welch, G., Towles, H.: Detailed Real-time Urban 3d Reconstruction from Video. *International Journal of Computer Vision* 78(23), 143–167 (2008)
12. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Reconstructing Building Interiors from Images. In: 12th International Conference on Computer Vision, pp. 80–87. IEEE Press, Kyoto (2009)
13. Golparvar-Fard, M., Savarese, S., Peña-Mora, F.: Interactive Visual Construction Progress Monitoring with 4D Augmented Reality Model. In: Construction Research Congress, Seattle, pp. 41–50 (2009)
14. Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S., Szeliski, R.: Reconstructing Rome. *IEEE Computer* 43(6), 40–47 (2010)
15. Gallup, D., Frahm, J., Pollefeys, M.: Piecewise Planar and Non-planar Stereo for Urban Scene Reconstruction. In: 23rd IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, pp. 1418–1425 (2010)
16. Bosché, F., Haas, C.T.: Automated Retrieval of 3D CAD Model Objects in Construction Range Images. *Automation in Construction* 17(4), 499–512 (2008)
17. Pu, S., George, V.: Knowledge Based Reconstruction of Building Models from Terrestrial Laser Scanning Data. *International Journal of Photogrammetry and Remote Sensing* 64(6), 575–584 (2009)
18. Tang, P., Huber, D., Akinci, B., Lipman, R., Lytle, A.: Automatic Reconstruction of As-built Building Information Models from Laser-scanned Point Clouds: A Review of Related Techniques. *Automation in Construction* 19(7), 829–843 (2010)
19. Son, H., Kim, C.: 3d Structural Component Recognition and Modeling Method using Color and 3d Data for Construction Progress Monitoring. *Automation in Construction* 19(7), 844–854 (2010)
20. Xiong, X., Huber, D.: Using Context to Create Semantic 3D Models of Indoor Environments. In: British Machine Vision Conference, Aberystwyth, pp. 45.1–45.11 (2010)
21. Huber, D., Akinci, B., Adan, O.A., Anil, E., Okorn, B.E., Xiong, X.: Methods for Automatically Modeling and Representing As-built Building Information Models. In: NSF Engineering Research and Innovation Conference, Atlanta (2011)
22. Adan, O.A., Xiong, X., Akinci, B., Huber, D.: Automatic Creation of Semantically Rich 3D Building Models from Laser Scanner Data. In: Proceedings of the International Symposium on Automation and Robotics in Construction (2011)
23. Valero, E.R., Adan, O.A., Huber, D., Cerrada, C.: Detection, Modeling, and Classification of Moldings for Automated Reverse Engineering of Buildings from 3D Data. In: 28th International Symposium on Automation and Robotics in Construction (2011)
24. VECO Project Technologies, <http://www.ch2m.com/corporate/markets/energy/veco.asp>
25. Reality Measurements Inc., <http://www.realitymeasurements.com>
26. Zhu, Z., Brilakis, I.: Comparison of Civil Infrastructure Optical-based Spatial Data Acquisition Techniques. *Journal of Computing in Civil Engineering* 23(3), 170–177 (2009)

27. Azhar, S., Hein, M., Sketo, B.: Building information modeling: Benefits, risks and challenges. In: 44th Associated Schools of Construction National Conference, Auburn (2008)
28. Nüchter, A., Surmann, H., Lingemann, K., Hertzberg, J.: Semantic scene analysis of scanned 3d indoor environments. In: Eighth International Fall Workshop on Vision, Modeling and Visualization, pp. 215–221 (2003)
29. Pu, S., George, V.: Knowledge based reconstruction of building models from terrestrial laser scanning data. *International Journal of Photogrammetry and Remote Sensing* 64(6), 575–584 (2009)
30. Pu, S.: Automatic building modeling from terrestrial laser scanning. In: Oosterom, P., Zlatanova, S., Penninga, F., Fendel, E.M., Cartwright, W., Gartner, G., Meng, L., Peterson, M.P. (eds.) *Advances in 3d Geoinformation Systems, Part II, Theme II. LNGC*, pp. 141–160. Springer, Heidelberg (2008)
31. Shin, S., Hryciw, R.D.: Wavelet Analysis of Soil Mass Images for Particle Size Determination. *Journal of Computing in Civil Engineering* 18(1), 19–27 (2004)
32. Masad, E., Al-Rousan, T., Button, J., Little, D., Tutumluer, E.: Test Methods for Characterizing Aggregate Shape, Texture, and Angularity. National Cooperative Highway Research Program (NCHRP), Report 555 (2007)
33. Pan, T., Tutumluer, E.: Imaging based evaluation of coarse aggregate size and shape properties affecting pavement performance. In: *Proceedings of Geo-Frontiers Congress, Austin* (2005)
34. Lee, S., Chang, L.M., Chen, P.H.: Performance comparison of bridge coating defect recognition method. *Corrosion* 61(1), 12–20 (2005)
35. Jeong, H., Abraham, D.M.: A decision tool for the selection of imaging technologies to detect underground infrastructure. *Tunneling and Underground Space Technology* 19(2), 175–191 (2003)
36. Hutchinson, T.C., Chen, Z.: Improved image analysis for evaluating concrete damage. *Journal of Computing in Civil Engineering* 20(3), 210–216 (2006)
37. Lester, J., Bernold, L.E.: Innovation to characterize buried utilities using Ground Penetrating Radar. *Automation in Construction* 16(4), 546–555 (2007)
38. Chen, Z.W., Xu, Y.L., Li, Q., Wu, D.J.: Dynamic Stress Analysis of Long Suspension Bridges under Wind, Railway, and Highway Loadings. *Journal of Bridge Engineering* 16, 383–392 (2008)
39. Chae, M.J., Iseley, T., Abraham, D.M.: Computerized sewer pipe condition assessment. In: *International Conference on Pipeline Engineering and Construction*, pp. 477–493. ASCE, Baltimore (2003)
40. Costello, S.B., Chapman, D.N., Rogers, C.D.F., Metje, N.: Underground asset location and condition assessment technologies. *Tunneling and Underground Space Technology* 22(5-6), 524–542 (2007)
41. Sinha, S.K., Fieguth, P.W.: Automated detection of Cracks in Buried Concrete Pipe Images. *Automation in Construction* 15(1), 58–72 (2006)
42. Yang, M.D., Su, T.C.: Segmenting ideal morphologies of sewer pipe defects on CCTV images for automated diagnosis. *Expert Systems with Applications* 36(2), 3562–3573 (2009)
43. Guo, W., Soibelman, L., Garrett, J.H.: Automated defect detection for sewer pipeline inspection and condition assessment. *Automation in Construction* 18(5), 587–596 (2009)
44. Zhu, Z., Brilakis, I.: Detecting Air Pockets for Architectural Concrete Quality Assessment using Visual Sensing. *Journal of Information Technology in Construction* 13, 86–102 (2008)

45. Zhu, Z., Brilakis, I.: Machine Vision based Concrete Surface Quality Assessment. *Journal of Construction Engineering and Management*, ASCE 136(2), 210–218 (2010)
46. Shih, N.J., Wu, M.C., Kunz, J.: The inspections of as-built construction records by 3D point clouds. Center for Integrated Facility Engineering, Working Paper #090, Stanford University (2004)
47. Akinci, B., Boukamp, F., Gordon, C., Huber, D., Lyons, C., Park, K.: A formalism for utilization of sensor systems and integrated project models for active construction quality control. *Automation in Construction* 15(2), 124–138 (2006)
48. Kim, C., Haas, C.T., Liapi, K.A.: Rapid, on-site spatial information acquisition and its use for infrastructure operation and maintenance. *Automation in Construction* 14, 666–684 (2005)
49. Kim, C., Son, H., Kim, H., Han, S.H.: Applicability of flash laser distance and ranging to three-dimensional spatial information acquisition and modeling on a construction site. *Canadian Journal of Civil Engineering* 35, 1331–1341 (2008)
50. Gong, J., Caldas, C.H.: Data processing for real-time construction site spatial modeling. *Automation in Construction* 17, 526–535 (2008)
51. Dai, F., Dong, S., Kamat, V.R., Lu, M.: Photogrammetry assisted measurement of interstory drift for rapid post-disaster building damage reconnaissance. *Journal of Nondestructive Evaluation* 30(3), 201–212 (2011)
52. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 519–526. IEEE Press (2006)
53. Furukawa, Y.: Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(8), 1362–1376 (2010)
54. Golparvar-Fard, M., Peña-Mora, F., Savarese, S.: D4AR – a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. *Journal of Information Technology in Construction* 14, 129–153 (2009)
55. Golparvar-Fard, M., Peña-Mora, F., Savarese, S.: D4AR – 4 dimensional augmented reality tools for automated remote progress tracking and support of decision-enabling tasks in the AEC/FM industry. In: *6th International Conference on Innovations in AEC* (2010)
56. Ibrahim, Y.M., Lukins, T.C., Zhang, X., Trucco, E., Kaka, A.P.: Towards automated progress assessment of workpackage components in construction projects using computer vision. *Advanced Engineering Informatics* 23, 93–103 (2009)
57. Quiñones-Rozo, C.A., Hashash, Y.M.A., Liu, L.Y.: Digital image reasoning for tracking excavation activities. *Automation in Construction* 17(5), 608–622 (2008)
58. González-Aguilera, D., Gómez-Lahoz, J.: Dimensional Analysis of Bridges from a Single Image. *Journal of Computing in Civil Engineering* 23(6), 319–329 (2009)
59. Dai, F., Lu, M.: Assessing the accuracy of applying photogrammetry to take geometric measurement on building products. *Journal of Construction Engineering and Management* 135(2), 242–250 (2010)
60. Chae, S., Kano, N.: Application of location information by stereo camera images to project progress monitoring. In: *24th International Symposium on Automation and Robotics in Construction*, Kochi, Kerala, India, pp. 89–92 (2007)
61. Tomasi, C., Kanade, T.: Detection and tracking of point features. *Carnegie Mellon University Technical Report* (1991)

62. Gupta, G., Balasubramanian, R., Rawat, M., Bhargava, R., Krishna, B.: Stereo matching for 3d building reconstruction. *Advances in Computing. Communication and Control* 125(3), 522–529 (2011)
63. Zhu, Z., Brilakis, I.: Concrete Column Recognition in Images and Videos. *Journal of Computing in Civil Engineering* 24(6), 478–487 (2010)
64. Yamaguchi, T., Hashimoto, S.: Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications* 11(5), 797–809 (2009)
65. Zhu, Z., German, S., Brilakis, I.: Visual Retrieval of Concrete Crack Properties for Automated Post-earthquake Structural Safety Evaluation. *Automation in Construction* 20(7), 874–883 (2011)
66. Zhu, Z., Brilakis, I.: Surface Defects Detection for Architectural Concrete Quality Assessment using Visual Sensing. *Special Issue in Sensors in Construction and Infrastructure Management. Journal of Information Technology in Construction* 13, 86–102 (2008)
67. German, S., Brilakis, I., DesRoches, R.: Automated Detection of Exposed Reinforcement in Post-Earthquake Safety and Structural Evaluations. In: *The 6th International Structural Engineering and Construction Conference* (2011)
68. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision* 43, 29–44 (2001)
69. Schmid, C.: Constructing models for content-based image retrieval. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 39–45 (2001)
70. Koch, C., Brilakis, I.: Pothole Detection in Asphalt Pavement Images. *Advanced Engineering Informatics* 25(3), 507–515 (2011)
71. Bouguet, J.Y.: Camera calibration toolbox for Matlab, Intel Corporation, [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
72. Fathi, H., Brilakis, I.: Automated sparse 3D point cloud generation of infrastructure using its distinctive visual features. *Advance Engineering Informatics* 25(4), 760–770 (2011)
73. Rashidi, A., Dai, F., Brilakis, I., Vela, P.: Comparison of camera motion estimation methods for 3D reconstruction of infrastructure. In: *2011 ASCE International Workshop on Computing in Civil Engineering* (2011)
74. Zhu, Z., Brilakis, I.: Concrete Column Recognition in Images and Videos. *Journal of Computing in Civil Engineering* 24(6), 478–487 (2010)
75. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: *IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE Press, Brazil (2007)
76. Savarese, S., Fei-Fei, L.: View Synthesis for Recognizing Unseen Poses of Object Classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III. LNCS*, vol. 5304, pp. 602–615. Springer, Heidelberg (2008)
77. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 778–785. IEEE Press, Miami (2009)
78. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 264–271 (2003)
79. Leibe, B., Schiele, B.: Scale-Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. In: Rasmussen, C.E., Bühlhoff, H.H., Schölkopf, B., Giese, M.A. (eds.) *DAGM 2004. LNCS*, vol. 3175, pp. 145–153. Springer, Heidelberg (2004)

80. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 524–531. IEEE Press (2005)
81. Felzenszwalb, P., Huttenlocher, D.: Efficient matching of pictorial structures. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 66–73. IEEE Press, South Carolina (2000)
82. Felzenszwalb, P., McAllester, D., Ramaman, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: 26th IEEE Conference on Computer Vision and Pattern Recognition (2008)
83. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 762–769 (2004)
84. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: 10th IEEE International Conference on Computer Vision, Beijing, vol. 2, pp. 1458–1465 (2005)
85. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding* 106, 59–70 (2007)
86. Schneiderman, H., Kanade, T.: A statistical approach to 3D object detection applied to faces and cars. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 746–751 (2000)
87. Weber, M., Welling, M., Perona, P.: Unsupervised Learning of Models for Recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 18–32. Springer, Heidelberg (2000)
88. Li, S.Z., Zhang, Z.: FloatBoost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(9), 1112–1123 (2004)
89. Brown, M., Lowe, D.G.: Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. In: 5th International Conference on 3-D Digital Imaging and Modeling, pp. 56–63. IEEE Press, Piscataway (2005)
90. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision* 67(2), 159–188 (2006)
91. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision* 66(3), 231–259 (2006)
92. Lowe, D.G.: Local feature view clustering for 3d object recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. I-682 – I-688. IEEE Press (2001)
93. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference, vol. 1, pp. 384–393 (2002)
94. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. *International Journal of Computer Vision*, 128–142 (2002)
95. Yan, P., Khan, D., Shah, M.: 3d model based object class detection in an arbitrary view. In: IEEE 11th International Conference on Computer Vision, pp. 1–6. IEEE Press, Rio de Janeiro (2007)
96. Thomas, A., Ferrar, V., Leibe, B., Tuytelaars, T., Schiel, B., Van Gool, L.: Towards multi-view object class detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1589–1596 (2006)



97. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3d feature maps. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, Anchorage (2008)
98. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3d object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, Minneapolis (2007)
99. Hoiem, D., Rother, C., Winn, J.: 3d layoutCRF for multi-view object class recognition and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, Minneapolis (2007)
100. Chiu, H., Kaelbling, L., Lozano-Perez, T.: Virtual training for multi-view object class recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, Minneapolis (2007)
101. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2036–2043. IEEE Press, Miami (2009)
102. 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: 12th International Conference on Computer Vision, pp. 213–220. IEEE Press, Kyoto (2009)
103. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3d object classes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1247–1254. IEEE Press, Miami (2009)
104. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
105. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference, pp. 147–151 (1988)
106. Tomasi, C., Kanade, T.: Detection and tracking of point features (1991)
107. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding* 110(3), 346–359 (2008)
108. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. *International Journal of Computer Vision* 60(1), 63–86 (2004)
109. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a Day. In: IEEE International Conference on Computer Vision, pp. 72–79. IEEE Press, Kyoto (2009)
110. Bok, Y., Choi, D., Jeong, Y., Kweon, I.S.: Capturing Village-Level Heritages with a Hand-Held Camera-Laser Fusion Sensor. In: 12th International Conference on Computer Vision Workshops (eHeritage and Digital Art Preservation), pp. 947–954. IEEE Press, Kyoto (2009)
111. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D.: The digital michelangelo project: 3D scanning of large statues. In: 27th Annual Conference on Computer Graphics and Iterative Techniques, pp. 131–144 (2000)
112. Bernardini, F., Martin, I.M., Rushmeier, H.: High-quality texture reconstruction from multiple scans. *IEEE Transactions of Visualization and Computer Graphics* 7(4), 318–332 (2001)
113. Shum, H.Y., Han, M., Szeliski, R.: Interactive construction of 3d models from panoramic mosaics. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 427–433. IEEE Press, Santa Barbara (1998)
114. Pollefeys, M., Van Gool, L.: From images to 3d models. *Communications of the ACM* 45(7), 50–55 (2002)

115. Dick, A.R., Torr, P.H.S., Cipolla, R.: Modeling and interpretation of architecture from several images. *International Journal of Computer Vision* 60(2), 111–134 (2004)
116. Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., Master, N.: Calibrated registered images of an extended urban area. *International Journal of Computer Vision* 53(1), 93–107 (2003)
117. Stamos, I., Allen, P.K.: Geometry and texture recovery of scene of large scale. *Journal of Computer Vision and Image Understanding* 88(2), 94–118 (2002)
118. Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., Dellaert, F.: Detecting and matching re-peated patterns for automatic geo-tagging in urban environments. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7. IEEE Press, Anchorage (2008)
119. Seitz, S., Dyer, C.: Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision* 35(2), 151–173 (1999)
120. Lindeberg, T.: Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics* 21(2), 224–270 (1994)

# Equi-affine Invariant Geometries of Articulated Objects

Dan Raviv, Alexander M. Bronstein, Michael M. Bronstein,  
Ron Kimmel, and Nir Sochen

Technion, Computer Science Department, Israel  
Tel Aviv University, School of Electrical Engineering, Israel  
Università della Svizzera Italiana, Faculty of Informatics, Switzerland  
Technion, Computer Science Department, Israel  
Tel Aviv University, Department of Applied Mathematics

**Abstract.** We introduce an (equi-)affine invariant geometric structure by which surfaces that go through squeeze and shear transformations can still be properly analyzed. The definition of an affine invariant metric enables us to evaluate a new form of geodesic distances and to construct an invariant Laplacian from which local and global diffusion geometry is constructed. Applications of the proposed framework demonstrate its power in generalizing and enriching the existing set of tools for shape analysis.

## 1 Introduction

Shape analysis has been one of the principal research fields in computer vision for many years. Numerous methods are based on modeling shapes as Riemannian manifolds, from which it is possible to derive many geometric invariances. Differential geometry and diffusion geometry have been bold players in this growing field. Schwartz *et al.* [22] proposed to embed a non-rigid shape in an Euclidean domain both conformal and isometric, followed by Elad *et al.* [14] that discussed embeddings in higher dimensions, and presented a practical representation of shapes referred to as *canonical forms*. Later on Elad *et al.* [13] and Bronstein *et al.* [5] showed that for some surfaces, such as faces, a spherical domain better captures intrinsic properties. In 2005 Memoli *et al.* [17] pointed the importance of *Gromov-Hausdorff* distance for shape analysis, followed by Bronstein *et al.* [6] who introduced a variational framework that minimizes the *Gromov-Hausdorff* distance by a direct embedding between two non-rigid shapes which does not suffer from an unbounded distortion of an intermediate ambient space. Diffusion geometry, referred to as spectral geometry, based on heat diffusion on manifolds and the properties of the Laplace Beltrami operator have become growingly popular in shape analysis in the past years. Drawing inspiration from Berard *et al.* 1994 work [2], Lafon *et al.* [10] proposed in 2006 a probabilistic analysis of algorithms using graph Laplacians. In 2007, Rustamov [21] showed how shapes can be analyzed using the eigen-functions of the Laplace Beltrami operator, and later on Gebal *et al.* [15] discussed auto diffusion functions. Sun *et al.* [24] used

the decay of heat as a feature, known as *Heat Kernel Signatures*, which was further used by [18] as volumetric descriptors. Diffusion geometric constructs in general were found to be more robust than their geodesic counterparts [7], hence they have found successful applications in many shape analysis tasks, such as [19].

However, all of these constructions depend on the definition of the Riemannian metric tensor. So far, the default choice of the metric induced by the Euclidean embedding of the shape has been used. Such a metric and all the related constructions is invariant to inelastic deformations of the shape and global Euclidean transformations (rotations, reflections and translations). In this paper, we show a different construction of a metric that has a wider class of invariance, being also invariant to equi-affine transformations. It contains the metric evaluation we presented in [29] and [30] for both diffusion and differential geometry.

The rest of the paper is organized as follows. In Section 2 we provide the mathematical background of Euclidean and diffusion geometry, followed by Section 3 where we elaborate on the equi-affine metric. Section 4 is dedicated to numerical aspects, and several applications are presented in Section 5. We conclude the paper in Section 6.

## 2 Mathematical Background

### 2.1 Differential Geometry

We model a surface  $(X, g)$  as a compact complete two dimensional Riemannian manifold  $X$  with a metric tensor  $g$ , evaluated on the tangent plane  $T_x X$  of point  $x$  in the natural basis using the inner product  $\langle \cdot, \cdot \rangle_x : T_x X \times T_x X \rightarrow \mathbb{R}$ . We further assume that  $X$  is embedded into  $\mathbb{E} = \mathbb{R}^3$  by means of a regular map  $\mathbf{x} : U \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , so that the metric tensor can be expressed in coordinates as

$$g_{ij} = \left\langle \frac{\partial \mathbf{x}}{\partial u_i}, \frac{\partial \mathbf{x}}{\partial u_j} \right\rangle, \quad (1)$$

where the  $u_i$ 's are the coordinates of  $U$ , which yields the infinitesimal displacement  $dp$

$$dp^2 = g_{11} du_1^2 + 2g_{12} du_1 du_2 + g_{22} du_2^2. \quad (2)$$

*Minimal geodesics*, or shortest paths, are the minimizers of all path length

$$d_X(x, x') = \min_{C \in \Gamma(x, x')} \ell(C) \quad (3)$$

over the set of all admissible paths  $\Gamma(x, x')$  between the points  $x$  and  $x'$  on the surface  $X$ , where due to completeness assumption, a minimizer always exists (not necessary unique). Many algorithms have been proposed for the computation of geodesic distances. They differ by accuracy and complexity. In this paper we focus on the family of algorithms simulating wavefront propagation known as *fast marching methods* [16].

### 2.2 Differential Operators

Laplace Beltrami operator (LBO), named after Eugenio Beltrami, is the generalization of the Laplace operator. It is a linear operator, defined as the divergence of the gradient of a scalar function  $f : X \rightarrow \mathbb{R}$  on a manifold

$$\Delta_g f = \operatorname{div}_g \operatorname{grad}_g f. \tag{4}$$

The operator can be extended to tensors, but it is beyond the scope of this note.

In local coordinates  $u$  of a chart [\[11\]](#), the LBO assumes the form of

$$\Delta_g f = \frac{1}{\sqrt{|g|}} \frac{\partial}{\partial u^\alpha} \left( \sqrt{|g|} g^{\alpha\beta} \frac{\partial}{\partial u^\beta} f \right), \tag{5}$$

where  $X(u^1, u^2, \dots, u^n) = (X^1, X^2, \dots, X^n)$  is the embedding of an  $n$ -dimensional manifold. Since our focus will be two dimensional affine invariants, we constrain ourself to two dimensions

$$X(u^1, u^2) = (x(u^1, u^2), y(u^1, u^2), z(u^1, u^2)). \tag{6}$$

### 2.3 Diffusion Geometry

The Laplace-Beltrami operator gives rise to the partial differential equation

$$\left( \frac{\partial}{\partial t} + \Delta_g \right) f(t, x) = 0, \tag{7}$$

called the *heat equation*. The heat equation describes the propagation of heat on the surface and its solution  $f(t, x)$  is the heat distribution at a point  $x$  in time  $t$ . The initial condition of the equation is some initial heat distribution  $f(0, x)$ ; if  $X$  has a boundary, appropriate boundary conditions must be added. The solution of [\(7\)](#) corresponding to a point initial condition  $f(0, x) = \delta(x - x')$ , is called the *heat kernel* and represents the amount of heat transferred from  $x$  to  $x'$  in time  $t$  by the diffusion process. Using spectral decomposition, the heat kernel can be represented as

$$h_t(x, x') = \sum_{i \geq 0} e^{-\lambda_i t} \phi_i(x) \phi_i(x') \tag{8}$$

where  $\phi_i$  and  $\lambda_i$  are, respectively, the eigenfunctions and eigenvalues of the Laplace-Beltrami operator satisfying  $\Delta \phi_i = \lambda_i \phi_i$  (without loss of generality, we assume  $\lambda_i$  to be sorted in increasing order starting with  $\lambda_0 = 0$ ). Since the Laplace-Beltrami operator is an *intrinsic* geometric quantity, i.e., it can be expressed solely in terms of the metric of  $X$ , its eigenfunctions and eigenvalues as well as the heat kernel are invariant under isometric transformations of the manifold.

The value of the heat kernel  $h_t(x, x')$  can be interpreted as the transition probability density of a random walk of length  $t$  from the point  $x$  to the point

$x'$ . This allows to construct a family of intrinsic metrics known as *diffusion metrics*,

$$d_t^2(x, x') = \int (h_t(x, y) - h_t(x', y))^2 dy = \sum_{i>0} e^{-\lambda_i t} (\phi_i(x) - \phi_i(x'))^2, \tag{9}$$

which measure the “connectivity rate” of the two points by paths of length  $t$ .

The parameter  $t$  can be given the meaning of *scale*, and the family  $\{d_t\}$  can be thought of as a scale-space of metrics. By integrating over all scales, a *scale-invariant* version of (9) is obtained,

$$d_{CT}^2(x, x') = 2 \int_0^\infty d_t^2(x, x') dt = \sum_{i>0} \frac{1}{\lambda_i} (\phi_i(x) - \phi_i(x'))^2. \tag{10}$$

This metric is referred to as the *commute-time distance* and can be interpreted as the connectivity rate by paths of any length. We will broadly call constructions related to the heat kernel, diffusion and commute time metrics as *diffusion geometry*.

### 3 Equi-affine Metric

An *affine transformation*  $\mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{b}$  of the three-dimensional Euclidean space can be parametrized by a regular  $3 \times 3$  matrix  $\mathbf{A}$  and a  $3 \times 1$  vector  $\mathbf{b}$ . since all constructions discussed here are trivially translation invariant, we will omit the vector  $\mathbf{b}$ . The transformation is called *special affine* or *equi-affine* if it is volume-preserving, i.e.,  $\det \mathbf{A} = 1$ .

As the standard Euclidean metric is not affine-invariant, the Laplace-Beltrami Operators associated with  $X$  and  $\mathbf{A}X$  are generally distinct, and so are the resulting diffusion geometries. In what follows, we are going to substitute the Euclidean metric by its equi-affine invariant counterpart. That, in turn, will induce an equi-affine-invariant Laplace-Beltrami Operator and define equi-affine-invariant diffusion geometry.

The equi-affine metric can be defined through the parametrization of a curve [8][23]. Let  $C$  be a curve on  $X$  parametrized by  $p$ . By the chain rule,

$$\begin{aligned} \frac{dC}{dp} &= \mathbf{x}_1 \frac{du_1}{dp} + \mathbf{x}_2 \frac{du_2}{dp} \\ \frac{d^2C}{dp^2} &= \mathbf{x}_1 \frac{d^2u_1}{dp^2} + \mathbf{x}_2 \frac{d^2u_2}{dp^2} + \mathbf{x}_{11} \left(\frac{du_1}{dp}\right)^2 + \\ &\quad 2\mathbf{x}_{12} \frac{du_1}{dp} \frac{du_2}{dp} + \mathbf{x}_{22} \left(\frac{du_2}{dp}\right)^2, \end{aligned} \tag{11}$$

where, for brevity, we denote  $\mathbf{x}_i = \frac{\partial \mathbf{x}}{\partial u_i}$  and  $\mathbf{x}_{ij} = \frac{\partial^2 \mathbf{x}}{\partial u_i \partial u_j}$ . As volumes are preserved under the equi-affine group of transformations, we define the invariant arclength  $p$  through

$$\det(\mathbf{x}_1, \mathbf{x}_2, C_{pp}) = 1. \quad (12)$$

Plugging (11) into (12) yields

$$dp^2 = \det(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_{11} du_1^2 + 2\mathbf{x}_{12} du_1 du_2 + \mathbf{x}_{22} du_2^2), \quad (13)$$

from where we readily have an equi-affine-invariant pre-metric tensor

$$\hat{g}_{ij} = \tilde{g}_{ij} |\tilde{g}|^{-1/4}, \quad (14)$$

where  $\tilde{g}_{ij} = \det(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_{ij})$ . The pre-metric tensor (14) defines a true metric only on strictly convex surfaces [8]; in more general cases, it might cease from being positive definite. In order to deal with arbitrary surfaces, we extend the metric definition by restricting the eigenvalues of the tensor to be positive. Representing  $\hat{g}$  as a  $2 \times 2$  matrix admitting the eigendecomposition  $\hat{\mathbf{G}} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T$ , where  $\mathbf{U}$  is orthonormal and  $\mathbf{\Gamma} = \text{diag}\{\gamma_1, \gamma_2\}$ , we compose a new first fundamental form for non-vanishing Gaussian curvature matrix  $\mathbf{G} = \mathbf{U}|\mathbf{\Gamma}|\mathbf{U}^T$ . The metric tensor  $g$  is positive definite and is equi-affine invariant.

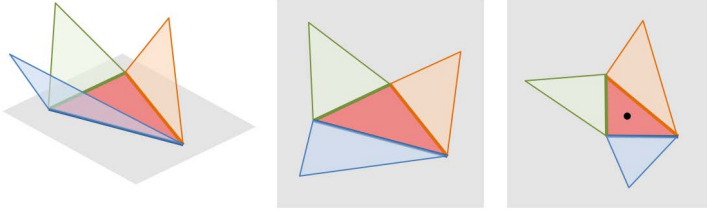
## 4 Numerical Considerations

### 4.1 Local Fitting

In order to compute the equi-affine metric we need to evaluate the second-order derivatives of the surface with respect to some parametrization coordinates. While this can be done practically in any representation, here we assume that the surface is given as a triangular mesh. For each triangular face, the metric tensor elements are calculated from a quadratic surface patch fitted to the triangle itself and its three adjacent neighbor triangles. The four triangles are unfolded to the plane, to which an affine transformation is applied in such a way that the central triangle becomes a unit simplex. The coordinates of this planar representation are used as the parametrization  $\mathbf{u}$  with respect to which the first fundamental form coefficients are computed at the barycenter of the simplex (Figure 1). This step is performed for every triangle of the mesh and is summarized in [30].

### 4.2 Affine Geodesics

Calculating geodesic distances was intensively explored in past decades. Several fast and accurate numerical schemes [27,16,25,26] can be used for this purpose. We use the FMM technique, after locally rescaling each edge according to the equi-affine metric. The (affine invariant) length of each edge is defined by  $L^2(dx, dy) = g_{11}dx^2 + 2g_{12}dxdy + g_{22}dy^2$ . Specifically, for our canonical triangle with vertices at  $(0, 0)$ ,  $(1, 0)$  and  $(0, 1)$  we have  $L_1^2 = g_{11}$ ,  $L_2^2 = g_{22}$  and  $L_3^2 = g_{11} - 2g_{12} + g_{22}$ . Each edge may appear in more than one triangle. In our experiments we use the average length as an approximation, while verifying that the triangle inequality holds.



**Fig. 1.** Left to right: part of a triangulated surface about a specific triangle. The three neighboring triangles together with the central one are unfolded flat to the plane. The central triangle is canonized into a right isosceles triangle; three neighboring triangles follow the same planar affine transformation. Finally, the six surface coordinate values at the vertices are used to interpolate a quadratic surface patch from which the metric tensor is computed.

### 4.3 Finite Elements Method (FEM)

Having the discretized first fundamental form coefficients, our next target is to discretize the Laplace-Beltrami Operator. Since our final goal is not the operator itself but its eigendecomposition, we skip the explicit construction of the Laplacian and discretize its eigenvalues and eigenfunctions directly. This is achieved using the finite elements method (FEM) proposed in [12] and used in shape analysis in [20]. For that purpose, we translate the eigendecomposition of the Laplace-Beltrami Operator  $\Delta\phi = \lambda\phi$  into a *weak form*

$$\int \psi_k \Delta\phi \, da = \lambda \int \psi_k \phi \, da \tag{15}$$

with respect to some basis  $\{\psi_k\}$  spanning a (sufficiently smooth) subspace of  $L^2(X)$ . Specifically, we choose the  $\psi_k$ 's to be the first-order finite element functions obtaining a value of one at a vertex  $k$  and decaying linearly to zero in its 1-ring (the size of the basis equals to the number of vertices in the mesh). Substituting these functions into (15), we obtain

$$\int \psi_k \Delta\phi \, da = \int \langle \nabla \psi_k, \nabla \phi \rangle_x \, da = \int g^{ij} (\partial_i \phi) (\partial_j \psi_k) \, da = \lambda \int \psi_k \phi \, da. \tag{16}$$

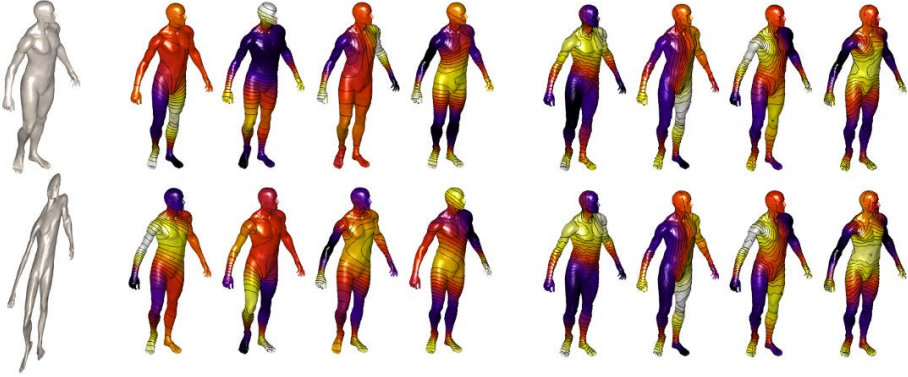
Next, we approximate the eigenfunction  $\phi$  in the finite element basis by  $\phi = \sum_{l=1} \alpha_l \psi_l$ . This yields

$$\int g^{ij} \left( \partial_i \sum_l \alpha_l \psi_l \right) (\partial_j \psi_k) \, da = \lambda \int \psi_k \sum_l \alpha_l \psi_l \, da,$$

or, equivalently,

$$\sum_l \alpha_l \int g^{ij} (\partial_i \psi_l) (\partial_j \psi_k) \, da = \lambda \sum_l \alpha_l \int \psi_k \psi_l \, da.$$





**Fig. 2.** Four eigenfunctions of the standard (second through fifth columns) and the proposed equi-affine-invariant (four rightmost columns) Laplace-Beltrami operator. Two rows show a shape and its equi-affine transformation. For convenience of visualization, eigenfunctions are textured mapped onto the original shape.

The last equation can be rewritten in matrix form as a generalized eigendecomposition problem  $\mathbf{A}\alpha = \lambda\mathbf{B}\alpha$  solved for the coefficients  $\alpha_l$ , where

$$a_{kl} = \int g^{ij} (\partial_i \psi_l) (\partial_j \psi_k) da,$$

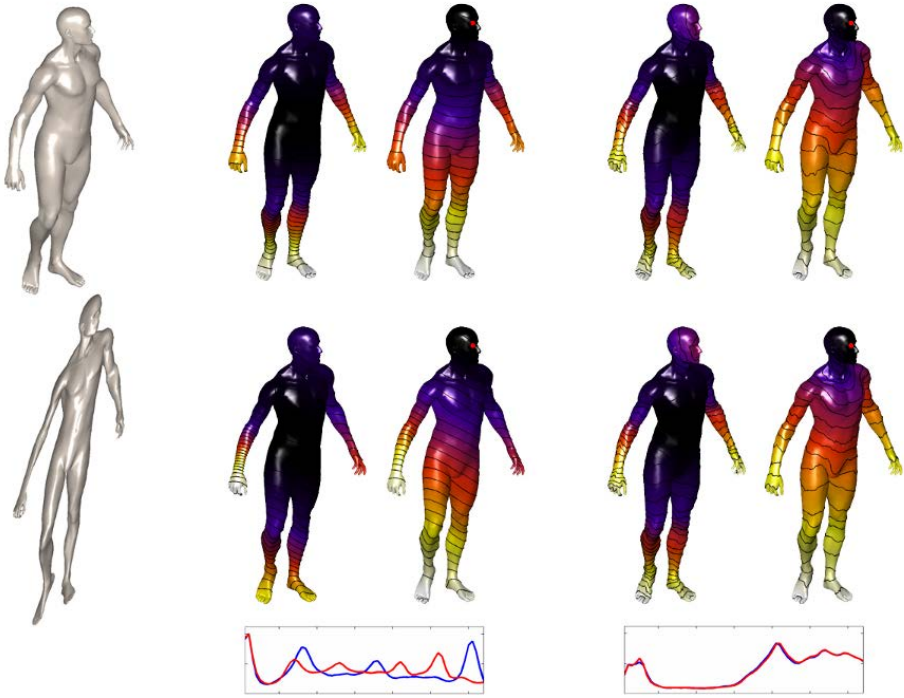
$$b_{kl} = \int \psi_k \psi_l da,$$

and the local surface area is expressed in parametrization coordinates as  $da = \sqrt{g} du_1 du_2$ . The resulting eigendecomposition can be used to define an equi-affine-invariant diffusion geometry. Eigenfunctions, heat kernels, and diffusion distances remain invariant under volume-preserving affine transformations of the shape (Figures 2-3).

Evaluating the proposed metric is bounded by the number of adjacent neighbors of each vertex, from which we conclude that the new metric is evaluated in linear time with relation of the number of vertices. Spectral decomposition is performed using the power method, implemented in MATLAB, and in practice we only need few (below 200) eigenvectors.

## 5 Applications

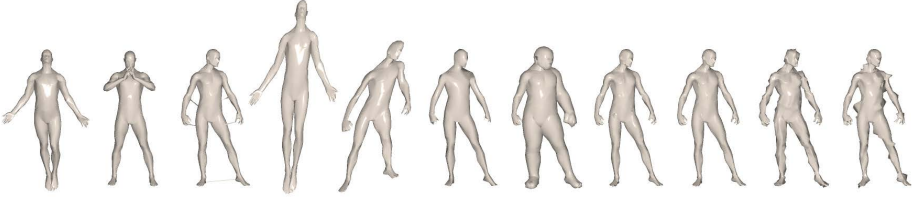
To evaluate the performance of the proposed approach for the construction of local descriptors, we used the Shape Google framework [28] based on standard and affine-invariant Heat Kernel Signatures. HKS and AI-HKS were computed at six arbitrary scales ( $t = 1024, 1351.2, 1782.9, 2352.5, \text{ and } 4096$ ). Bags of features were computed using soft vector quantization with variance taken as twice the median of all distances between cluster centers. Approximate nearest neighbor method [1] was used for vector quantization. Both the standard and the affine-invariant Laplace-Beltrami Operator discretization were computed using finite



**Fig. 3.** Heat kernel signature  $h_t(x, x)$  and diffusion metric ball (second and third columns, respectively), and their equi-affine invariant counterparts (fourth and fifth columns, respectively). Two rows show a shape and its transformation. For convenience of visualization, the kernel and the metric are overlaid onto the original shape. Plots under the figure show the corresponding metric distributions before and after the transformation.

elements. Heat kernels were approximated using the first 100 eigenpairs of the discrete Laplacian. The geometric vocabulary size was set to 64.

Evaluation was performed using the SHREC 2010 robust large-scale shape retrieval benchmark methodology [4]. The dataset consisted of two parts: 793 shapes from 13 shape classes with simulated transformation of different types (Figure 4) and strengths (60 per shape) used as queries, and additional 521 shapes from a large variety of objects. The total dataset size was 1314. Retrieval was performed by matching 780 transformed queries to shape classes. Each query had one correct corresponding null shape in the dataset. Performance was evaluated using precision/recall characteristic. *Precision*  $P(r)$  is defined as the percentage of relevant shapes in the first  $r$  top-ranked retrieved shapes. *Mean average precision* (mAP), defined as  $mAP = \sum_r P(r) \cdot rel(r)$ , where  $rel(r)$  is the relevance of a given rank, was used as a single measure of performance. Intuitively, mAP is interpreted as the area below the precision-recall curve. Ideal performance retrieval performance results in first relevant match with mAP=100%. Performance results were broken down according to transformation class and strength.



**Fig. 4.** Examples of query shape transformations used in the shape retrieval experiment (left to right): null, isometry, topology, affine, affine+isometry, sampling, local dilation, holes, microholes, Gaussian noise, shot noise

Tables 2-11 show that in contrast to the Euclidean metric, the equi-affine metric preserves the high accuracy rate of shape retrieval for all deformations, including equi-affine. In some deformations we can see an improvement, which we attribute to the smoothing effect of the second order interpolation. As this metric is based on second derivatives it is less robust to noise than its Euclidean adversary. Yet, since the numeric is based on the weak form (FEM) of the LBO, the integration improves robustness. Adding that to the usage of low frequencies from the eigendecomposition, explains the competitive results even without performing noise reduction and/or resampling as a preprocessing step.

The equi-affine metric can be used in many existing methods that compute geodesic distances. In what follows, we show several examples for using the new metric in known applications such as Voronoi tessellation and non-rigid matching.

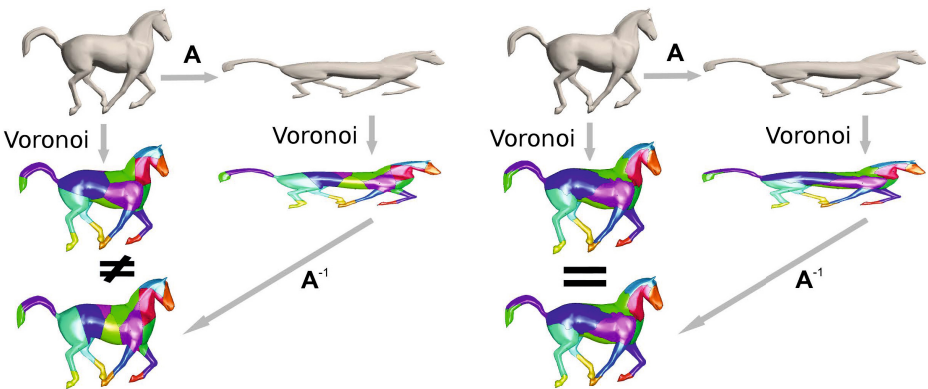
Voronoi tessellation is a partitioning of  $(X, g)$  into disjoint open sets called Voronoi cells. A set of  $k$  points  $(x_i \in X)_{i=1}^k$  on the surface defines the Voronoi cells  $(V_i)_{i=1}^k$  such that the  $i$ -th cell contains all points in  $X$  closer to  $x_i$  than to any other  $x_j$  in the sense of the metric  $g$ . Voronoi tessellations created with the equi-affine metric commute with equi-affine transformations as visualized in Figure 5.

**Table 1.** Performance (mAP in %) of Shape Google with HKS descriptors

Transform.	Strength				
	1	≤2	≤3	≤4	≤5
<i>Isometry</i>	100.00	100.00	100.00	100.00	100.00
<i>Equi-Affine</i>	100.00	86.89	73.50	57.66	46.64
<i>Iso.+Equi-Affine</i>	94.23	86.35	76.84	70.76	65.36
<i>Topology</i>	100.00	100.00	98.72	98.08	97.69
<i>Holes</i>	100.00	96.15	92.82	88.51	82.74
<i>Micro holes</i>	100.00	100.00	100.00	100.00	100.00
<i>Local scale</i>	100.00	100.00	97.44	87.88	78.78
<i>Sampling</i>	100.00	100.00	100.00	96.25	91.43
<i>Noise</i>	100.00	100.00	100.00	99.04	99.23
<i>Shot noise</i>	100.00	100.00	100.00	98.46	98.77

**Table 2.** Performance (mAP in %) of Shape Google with equi-affine-invariant HKS descriptors

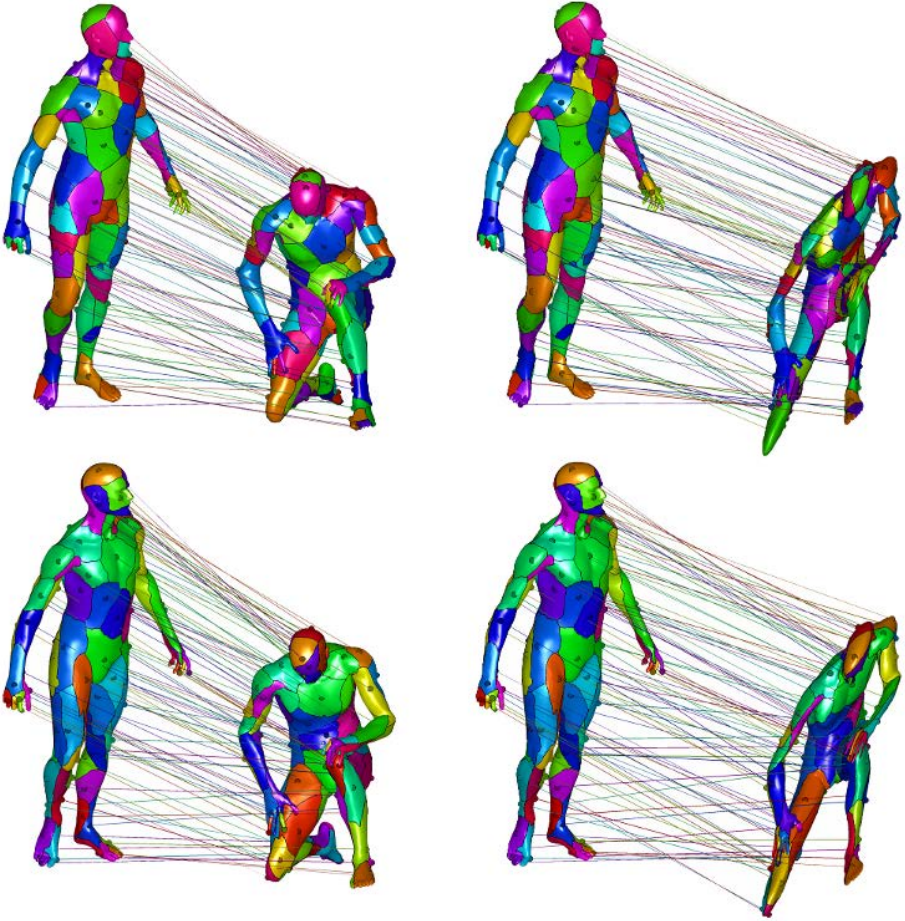
Transform.	Strength				
	1	≤2	≤3	≤4	≤5
<i>Isometry</i>	100.00	100.00	100.00	100.00	99.23
<i>Affine</i>	100.00	100.00	100.00	100.00	97.44
<b><i>Iso. + Equi-Affine</i></b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
<i>Topology</i>	96.15	94.23	91.88	89.74	86.79
<i>Holes</i>	100.00	100.00	100.00	100.00	100.00
<i>Micro holes</i>	100.00	100.00	100.00	100.00	100.00
<i>Local scale</i>	100.00	100.00	94.74	82.39	73.97
<i>Sampling</i>	100.00	100.00	100.00	96.79	86.10
<i>Noise</i>	100.00	100.00	89.83	78.53	69.22
<i>Shot noise</i>	100.00	100.00	100.00	97.76	89.63

**Fig. 5.** Voronoi cells generated by a fixed set of 20 points on a shape undergoing an equi-affine transformation. The standard geodesic metric (left) and its equi-affine counterpart (right) were used. Note that in the latter case the tessellation commutes with the transformation.

Two non-rigid shapes  $X, Y$  can be considered similar if there exists an isometric correspondence  $\mathcal{C} \subset X \times Y$  between them, such that  $\forall x \in X$  there exists  $y \in Y$  with  $(x, y) \in \mathcal{C}$  and vice-versa, and  $d_X(x, x') = d_Y(y, y')$  for all  $(x, y), (x', y') \in \mathcal{C}$ , where  $d_X, d_Y$  are geodesic distance metrics on  $X, Y$ . In practice, no shapes are perfectly isometric, and such a correspondence rarely exists; however, one can attempt finding a correspondence minimizing the metric *distortion*,

$$\text{dis}(\mathcal{C}) = \max_{\substack{(x,y) \in \mathcal{C} \\ (x',y') \in \mathcal{C}}} |d_X(x, x') - d_Y(y, y')|. \quad (17)$$

The smallest achievable value of the distortion is called the *Gromov-Hausdorff distance* [9] between the metric spaces  $(X, d_X)$  and  $(Y, d_Y)$ ,



**Fig. 6.** The GMDS framework is used to calculate correspondences between a shape and its isometry (left) and isometry followed by an equi-affine transformation (right). Matches between shapes are depicted as identically colored Voronoi cells. Standard distance (first row) and its equi-affine-invariant counterpart (second row) are used as the metric structure in the GMDS algorithm. Inaccuracies obtained in the first case are especially visible in the legs and arms.

$$d_{\text{GH}}(X, Y) = \frac{1}{2} \inf_C \text{dis}(C), \quad (18)$$

and can be used as a criterion of shape similarity.

The choice of the distance metrics  $d_X, d_Y$  defines the invariance class of this similarity criterion. Using geodesic distances, the similarity is invariant to inelastic deformations. Here, we use geodesic distances induced by our equi-affine Riemannian metric tensor, which gives additional invariance to affine transformations of the shape.

Bronstein *et al.* [3] showed how (18) can be efficiently approximated using a convex optimization algorithm in the spirit of multidimensional scaling (MDS), referred to as generalized MDS (GMDS). Since the input of this numeric framework are geodesic distances between mesh points, all that is needed to obtain an equi-affine GMDS is one additional step where we substitute the geodesic distances with their equi-affine equivalents. Figure 6 shows the correspondences obtained between an equi-affine transformation of a shape using the standard and the equi-affine-invariant versions of the geodesic metric.

## 6 Conclusion

We introduced an equi-affine-invariant metric that can cope with surfaces that do not have vanishing Gaussian curvature. We showed a wide range of applications, from shape retrieval through Voronoi tessellation to correspondence search, based on differential geometry tools and spectral analysis. The limitation of the method is the fixed scale restriction that will be solved in the future.

**Acknowledgments.** This research was supported by European Community's FP7- ERC program, grant agreement no. 267414. MB was supported by the Swiss High-Performance and High-Productivity Computing (HP2C).

## References

1. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching. *J. ACM* 45, 891–923 (1998)
2. Bérard, P., Besson, G., Gallot, S.: Embedding riemannian manifolds by their heat kernel. *Geometric and Functional Analysis* 4, 373–398 (1994)
3. Bronstein, A., Bronstein, M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing* 28(5), 1812–1836 (2006)
4. Bronstein, A.M., Bronstein, M.M., Castellani, U., Falcidieno, B., Fusiello, A., Godil, A., Guibas, L.J., Kokkinos, I., Lian, Z., Ovsjanikov, M., Patané, G., Spagnuolo, M., Toldo, R.: SHREC 2010: robust large-scale shape retrieval benchmark. In: *Proc. 3DOR* (2010)
5. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Expression-invariant face recognition via spherical embedding. In: *Proc. Int'l Conf. Image Processing (ICIP)*, vol. 3, pp. 756–759 (2005)
6. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing* 28, 1812–1836 (2006)
7. Bronstein, M.M., Bronstein, A.M.: Shape recognition with spectral distances with spectral distances. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 33, 1065–1071 (2011)

8. Buchin, S.: Affine differential geometry. Science Press, Beijing (1983)
9. Burago, D., Burago, Y., Ivanov, S.: A course in metric geometry. Graduate studies in mathematics, vol. 33. American Mathematical Society (2001)
10. Coifman, R.R., Lafon, S.: Diffusion maps. *Applied and Computational Harmonic Analysis* 21, 5–30 (2006)
11. Dierkes, U., Hildebrandt, S., Kuster, A., Wohlrab, O.: *Minimal Surfaces I*. Springer, Heidelberg (1992)
12. Dziuk, G.: Finite elements for the Beltrami operator on arbitrary surfaces. In: Hildebrandt, S., Leis, R. (eds.) *Partial Differential Equations and Calculus of Variations*, pp. 142–155 (1988)
13. Elad, A., Keller, Y., Kimmel, R.: Texture mapping via spherical multi-dimensional scaling. In: *Proc. Scale-Space Theories in Computer Vision*, pp. 443–455 (2005)
14. Elad, A., Kimmel, R.: On bending invariant signatures for surfaces. *Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 25, 1285–1295 (2003)
15. Gebal, K., Bærentzen, J.A., Aanæs, H., Larsen, R.: Shape analysis using the auto diffusion function. In: *Proc. of the Symposium on Geometry Processing*, pp. 1405–1413 (2009)
16. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. *Proc. National Academy of Sciences (PNAS)* 95, 8431–8435 (1998)
17. Mémoli, F., Sapiro, G.: A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* 5, 313–346 (2005)
18. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Volumetric heat kernel signatures. In: *Proc. 3D Object recognition (3DOR)*, part of ACM Multimedia (2010)
19. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Sapiro, G.: Diffusion symmetries of non-rigid shapes. In: *Proc. International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2010)
20. Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., Spagnuolo, M.: Discrete Laplace–Beltrami operators for shape analysis and segmentation. *Computers & Graphics* 33, 381–390 (2009)
21. Rustamov, R.M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In: *Proc. Symposium on Geometry Processing (SGP)*, pp. 225–233 (2007)
22. Schwartz, E.L., Shaw, A., Wolfson, E.: A numerical solution to the generalized mapmaker’s problem: flattening nonconvex polyhedral surfaces. *Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 11, 1005–1008 (1989)
23. Sochen, N.: Affine-invariant flows in the Beltrami framework. *Journal of Mathematical Imaging and Vision* 20, 133–146 (2004)
24. Sun, J., Ovsjanikov, M., Guibas, L.J.: A concise and provably informative multi-scale signature based on heat diffusion. In: *Proc. Symposium on Geometry Processing (SGP)* (2009)
25. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S., Hoppe, H.: Fast exact and approximate geodesics on meshes. In: *Proc. ACM Transactions on Graphics (SIGGRAPH)*, pp. 553–560 (2005)
26. Yatziv, L., Bartesaghi, A., Sapiro, G.:  $O(N)$  implementation of the fast marching algorithm. *J. Computational Physics* 212, 393–399 (2006)

27. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Trans. Automatic Control* 40, 1528–1538 (1995)
28. Ovsjanikov, M., Bronstein, A.M., Bronstein, M.M., Guibas, L.J.: Shape Google: a computer vision approach to invariant shape retrieval. In: *Proc. NORDIA* (2009)
29. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Sochen, N.: Affine-invariant diffusion geometry of deformable 3D shapes. In: *Proc. Computer Vision and Pattern Recognition (CVPR)* (2011)
30. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Sochen, N.: Affine-invariant geodesic geometry of deformable 3D shapes. *Computers & Graphics* 35, 692–697 (2011)



# Towards Fast Image-Based Localization on a City-Scale

Torsten Sattler<sup>1</sup>, Bastian Leibe<sup>2</sup>, and Leif Kobbelt<sup>1</sup>

<sup>1</sup> RWTH Aachen University, Aachen, Germany  
{tsattler,kobbelt}@cs.rwth-aachen.de

<sup>2</sup> UMIC Research Centre, RWTH Aachen University, Aachen, Germany  
leibe@umic.rwth-aachen.de

**Abstract.** Recent developments in Structure-from-Motion approaches allow the reconstructions of large parts of urban scenes. The available models can in turn be used for accurate image-based localization via pose estimation from 2D-to-3D correspondences. In this paper, we analyze a recently proposed localization method that achieves state-of-the-art localization performance using a visual vocabulary quantization for efficient 2D-to-3D correspondence search. We show that using only a subset of the original models allows the method to achieve a similar localization performance. While this gain can come at additional computational cost depending on the dataset, the reduced model requires significantly less memory, allowing the method to handle even larger datasets. We study how the size of the subset, as well as the quantization, affect both the search for matches and the time needed by RANSAC for pose estimation.

**Keywords:** image localization, image retrieval, camera pose estimation.

## 1 Introduction

Image-based localization methods try to estimate the position from which a query image was taken. Once obtained, the position can be used to determine, e.g., the position of a pedestrian [20,29,40,7] or of a mobile robot [11,12,37]. An especially interesting application is image-based localization for mobile devices, where a user simply sends a photo taken with her mobile phone to a server and in return receives information about her position [7]. Camera positions computed by localization methods are also useful for Structure-from-Motion reconstructions [1,10,14,18,28,35] or for the visualization of photo collections [34].

In order to enable image-based localization, some kind of visual representation of the scene is required. Traditionally, the chosen representation has been a set of images, enabling the use of image retrieval methods to efficiently find similar images and then use the (GPS) positions of the images as an approximation to the position of the query camera. Such a representation usually contains a lot of redundant information as multiple images cover the same part of the scene. Furthermore, many confusing features found in the images have to be removed for better retrieval [22]. The redundancy in the image set can be exploited to obtain

a Structure-from-Motion (SfM) reconstruction of the scene [1,10,14,18,28,35], resulting in a 3D point cloud. The image matching part of the SfM pipeline automatically removes most of the confusing features. Thus, a 3D reconstruction offers a more compact representation of the scene than the original images.

While a purely image-based representation only allows to compute the position of the camera, using a 3D model to represent the scene offers the additional advantage that the full camera pose, i.e., both position and orientation, can be determined. Essential for camera pose estimation are correspondences between 2D features in the query image and 3D points in the model. For every 3D point there is a list of 2D image features obtained from the images used to triangulate the point. These features model the appearance of the point from multiple viewpoints under varying lighting conditions. By also extracting local features in the query image, the correspondence search can be modeled as a descriptor matching problem. Due to the large scale of the reconstructions, containing one million or more points, the search method needs to be efficient. Additionally, it has to find enough correspondences to allow pose estimation. Furthermore, most of the matches have to be correct to avoid spending too much time on RANSAC.

A common approach for fast correspondence search is to first find an intermediate representation to quickly narrow down the search for possible correspondence, for example by only considering points found in database images similar to the query image [20]. Recently, Sattler et al. showed that direct search approaches that consider all 3D points with similar enough descriptors as potential correspondences for a feature in the query image achieve a better localization performance, i.e., are able to localize more images [30]. They propose a direct search method based on a visual vocabulary which limits the correspondence search of a query feature to all 3D points with descriptors assigned to the same visual word. Combined with a prioritization scheme, their approach is able to outperform current state-of-the-art methods either in localization performance or efficiency or both. In this paper, we look at two aspects of the method that are critical for scalability to larger datasets: First, the method requires to keep multiple descriptors for every 3D point in memory for efficient nearest neighbor search. Second, as more 3D points are used, the space containing the descriptors becomes denser. As the method uses SIFT features together with SIFT ratio-test [24] to reject wrong correspondences, a denser search space will most likely remove more correct correspondences as well. A simple way to reduce the memory footprint is to use only a subset of the 3D points available in the model. Using fewer points, and thus fewer descriptors, can also have a positive effect on the localization performance for larger datasets if the descriptor space also becomes sparser. In this paper, we experimentally evaluate the impact of considering subsets of the points in the model, selected by a simple reduction scheme recently proposed by Li et al. [23]. More specifically, we explore the relation between the number of points used, localization performance and efficiency, as well as localization accuracy. We show that we can achieve a similar registration performance at comparable efficiency and slightly better accuracy when using less than half of the points originally contained in the model. To explore the effect of using fewer points on the descriptor space, we simulate a larger dataset by

combining multiple smaller ones. Our experiments show that using subsets of the points cannot prevent the descriptor space from becoming too dense, but can speed up the registration process while maintaining a similar registration performance.

We use the notation introduced in [23], referring to 2D local features found in images and their descriptors as *features* and to 3D points and their descriptors from the database images as *points*. A *visual vocabulary* is obtained by clustering a set of local features using approximate k-means [27]. The obtained cluster centers are called *visual words*. Assigning a feature to its visual words means finding the cluster center which has the closest Euclidean distance to the feature through approximate nearest neighbor search.

The paper is structured as follows. Section 2 reviews related work. Section 3 discusses the approaches from [23,30] in more detail as they are the most relevant work to the work presented in this paper. We experimentally evaluate the combination of the method from [30] and the point filtering proposed in [23] in Section 4. Section 5 concludes the paper by discussing future work.

## 2 Related Work

Robertson and Cipolla developed one of the earliest image-based methods for localization. Their database consists of 200 image of facades in an urban environment, which are rectified to allow invariance against viewpoint changes [29]. The approach of Zhang & Kosecka retrieves the two images in a database that are most similar to a given query image [40], but instead of canonic views they use SIFT features to handle viewpoint differences. The position of the query camera is then triangulated from the GPS positions of the two retrieved images. Schindler et al. use 30k images, each one associated to a GPS position, to model large parts of a city [31]. To scale their localization method to such a large dataset, they accelerate the image retrieval step through the vocabulary tree method developed by Nister and Stewenius [26], using only features that are informative about their location to obtain a discriminative vocabulary. While Schindler et al. operate on a visual word level, Zamir and Shah use the original SIFT descriptors found in 100k database images, storing the descriptors in a tree-structure [39]. They propose an adapted SIFT ratio-test to deal with repetitive features and achieve positional accuracy comparable to GPS using a voting scheme. To handle an ever larger dataset of around 1 million images, Avrithis et al. aggregate the information of multiple images depicting the same scene into scene maps [4]. This clustering has the positive effect that it increases the recall while reducing the number of documents in the database. A still larger, planet-scale level with more than 6 million database images is considered by Hays and Efros who achieve localization through finding the modes of a probability distribution of possible locations all over the globe [19].

In robotics, the scene in which a robot operates might not be known in advance. In this case, cameras mounted on the robot are used to build a 3D reconstruction of the environment. This model is in turn used to estimate the relative position and orientation of the robot. An early version of such a simultaneous localization and mapping (SLAM) system has been proposed by Se et al.

[32]. Current state-of-the-art methods such as [6,11,12] try to adapt the SLAM approach to increasingly large scenes for real-time localization.

For large scenes, the construction of the 3D model cannot be achieved in real-time anymore. In case of a static environment the reconstruction can be precomputed using Structure-from-Motion techniques. Irschara et al. propose an approach that uses such models for image-based localization [20]. To narrow down the set of points that have to be considered to establish 2D-to-3D correspondences, they use an image retrieval step to find similar images from the set of images used for the reconstruction. Efficient GPU implementations for both feature matching and vocabulary tree-based retrieval enable their approach to perform in real-time. In order to localize query images substantially different from the database images, Irschara et al. place synthetic cameras on the ground plane to generate additional views. A informative subset of images is picked from the set of original and new images to form the database for retrieval. Wendel et al. generalize the placement of virtual cameras to full 3D to use a similar pipeline for the localization of aerial vehicles [37]. In another retrieval-based approach, Arth et al. use manually selected 3D point sets together with the images the points are visible in for pose estimation on mobile phones [2].

Li et al. show that directly establishing 3D-to-2D correspondences without the intermediate image retrieval step improves localization performance [23]. Starting with points visible in many database images, their prioritized matching algorithm tries to match 3D points to the 2D features in the query image. A point selection schemes computes a more compact representation of the original reconstruction. They show that using such a reduced model improves both localization performance and registration time. Sattler et al. present another approach that directly tries to establish correspondences [30]. In contrast to Li et al. they perform 2D-to-3D matching of 2D features against 3D points. To accelerate the correspondence search they use a prioritization scheme that first evaluates features for which only a small part of the descriptor space has to be searched. The search cost associated with each feature is estimated using a quantization of descriptor space defined by a visual vocabulary.

### 3 Prioritized Search

In this paper we evaluate the combination of the localization method from Sattler et al. [30] with the point selection scheme from Li et al. [23], aiming to achieve a similar localization performance and efficiency using fewer points and thus less memory. In the following, we review the two approaches.

Both methods are based on the key observation that not all 2D-to-3D correspondences that can be found are needed to successfully estimate the camera pose. The search time can be reduced by applying a prioritization scheme that first considers the most promising features and stops the search if enough correspondences are found. As Li et al. and Sattler et al. perform matching in opposite directions, their prioritization schemes are fundamentally different.

Li et al. try to match 3D points against 2D image features (3D-to-2D matching). They establish a correspondence between a 3D point with mean descriptor

$d$  and a 2D feature with descriptor  $d_1$  if the SIFT ratio-test  $\frac{\|d-d_1\|_2}{\|d-d_2\|_2} < 0.7$  is fulfilled. Here  $d_1$  and  $d_2$  are the first and second nearest neighbors for  $d$  amongst the descriptors in the query image, found through approximate tree-based search [3]. Their prioritization scheme is based on co-visibility of points since a match found for a point  $p$  increases the likelihood to find a correspondence for points visible together with  $p$ . To this end, two points  $p$  and  $p'$  are considered to be visible together if there is at least one database image that contains both points. The initial priority of a point is related to the number of database images it is visible in. In case the model was constructed from images obtained from a photo-sharing website, the method thus favors points visible in regions where many photos were taken, i.e., regions which seem to be interesting for tourists. If the model was built from more evenly distributed images, e.g., street view panoramas, stable points visible under different viewing angles are preferred. When a correspondence for  $p$  is found the priority of a point  $p'$  is increased if  $p$  and  $p'$  are visible together. The search for correspondences is stopped as soon as  $N_t = 100$  correspondences are found. Observing that about one out of every 500 point creates a correspondences by pure chance, Li et al. stop the search as soon as  $500 \cdot N_t = 50,000$  points have been considered [23].

Large-scale reconstructions contain millions of 3D points and some query images might see only 3D points whose priority is so low that the search would be stopped before any of them are considered. To circumvent this problem, Li et al. propose to use a set of *seed points* [23] that contains locally important points from all over the model. By giving these points a higher priority than all other 3D points, they perform a breadth-first search on the set of seed points to quickly converge to the area of the model that is likely to be seen in the image [23]. The set of seed points is computed by solving a set cover problem, where every point covers all images it is visible in. The seed set is constructed by finding a (minimal) set of points such that every image in the database is covered by at least 5 points. Since computing the minimum set cover is NP-hard, Li et al. use a simple greedy algorithm that iteratively selects the point that covers the largest number of images that have not yet been covered by 5 points [23]. The greedy algorithm is stopped after finding 2000 points to keep the set of seed points compact. Li et al. also use a compact model, again obtained from the greedy algorithm ensuring that every image is covered by at least  $K$  points without any limit on the number of selected points, instead of the full 3D model containing all points. They show experimentally that 3-20% of the original features (depending on the structure of the dataset) suffice to achieve both faster localization times and better localization performance, as more images can be registered using the reduced model than with the original model.

While Li et al. match points against features, Sattler et al. propose an approach that performs matching in the other direction (2D-to-3D matching) [30]. They observe that a simple method that stores the mean descriptor for every 3D point in a kd-tree and then performs approximate search [25] for the two nearest neighbors for every query feature, followed by applying the SIFT ratio-test and RANSAC-based pose estimation, achieves better localization performance than

current state-of-the-art methods [23]. While offering excellent performance, this method is way too slow for practical applications. Sattler et al. argue that this is due to wasting most of the search time on features that have no correspondence to 3D points in the scene. Instead of treating every feature the same way, they propose a prioritization scheme that first evaluates features for which one can quickly decide whether they lead to a correspondence or not. The cost of matching a 2D feature against the reconstruction is related to the number of points that have to be considered. To simultaneously limit the search space and estimate the search cost, Sattler et al. quantize the descriptor space of the used SIFT features into visual words using approx. k-means clustering [27]. In an offline process, the descriptors of the 3D points are assigned to visual words and for each word the list of points that have at least one descriptor assigned to it is stored together with the corresponding feature descriptors. Considering only the points assigned to the same visual word allows to relate the search cost of a query feature to the number of points stored in its word.

Given a new query image and the local features extracted from it, the localization method by Sattler et al. first assigns every feature in the image to its nearest visual word using approximate kd-tree search [25]. The list of (feature,word) pairs is then sorted in increasing number of (point,descriptor) pairs assigned to the words during the offline process. The features in the image are inspected in this order. Given the currently considered feature  $f$ , the method performs a linear search through all (point,descriptor) pairs stored in the visual word the descriptor  $d_f$  of  $f$  was assigned to. The search finds the two points  $p$ ,  $q$  ( $p \neq q$ ) whose descriptors  $d_p$ ,  $d_q$  are the nearest neighbors of  $d_f$ . Similar to [23], a correspondence between the feature  $f$  and the point  $p$  is established if the SIFT ratio-test  $\frac{\|d_f - d_p\|_2}{\|d_f - d_q\|_2} < 0.7$  is fulfilled. Since the 3D model is obtained from a SfM reconstruction, every point has at least two descriptors assigned to it. Therefore, a point can potentially be assigned to multiple visual words. To avoid establishing multiple correspondences containing the same 3D point, a newly found correspondence  $(f', p)$  replaces an existing correspondence  $(f, p)$  if  $\|d_{f'} - d_p\|_2 < \|d_f - d_p\|_2$  and is rejected otherwise. The search for further correspondences is stopped when  $N_t$  correspondences are found. Similar to Li et al., the 6-point DLT algorithm [17] is used to estimate the camera pose inside a RANSAC [13] loop. For robust estimation, a randomized RANSAC variant [9] is used in conjunction with a local optimization scheme [8].

Sattler et al. rigorously explore the design space of this method through experiments on the datasets from [20,23], showing that their method outperforms other state-of-the-art methods such as [20,23] in either localization performance or efficiency or even both. They explore different strategies to represent 3D points by their descriptors, reporting that the following two give the best results: The *all descriptors* (all desc.) strategy represents every 3D point by all of its descriptors. As a result, more than one descriptor of a point can be stored in the same visual word, increasing the search time for the word. The *integer mean per visual word* (int. mean) strategy tries to reduce the memory requirements by replacing multiple descriptors of the same point assigned to the same word by their mean

descriptor. The entries of this mean descriptor are then rounded to the nearest integer value to be able to use only 1 byte for each entry instead of the 4 bytes needed by a floating point representation. Choosing  $N_t = 100$  helps to reduce the search times without any significant negative impact on the registration performance. Furthermore, assuming an initial inlier ratio of 20% for RANSAC effectively limits the maximal number of taken samples with little impact on the localization performance. The source code for the method has been made publicly available and can be found at <http://www.graphics.rwth-aachen.de/localization/>.

There is an interesting analogy between the prioritization scheme of Sattler et al. and the well-known idf-weighting scheme used in image retrieval [33]. The idf-scheme weights down words that are used in many documents since they are less discriminative. Similarly, the prioritization scheme from [30] favors features mapped to a visual word which does not occur very often in the model and thus contains discriminative points. Therefore, besides trying to minimize the search costs by finding a suitable ordering of features, the prioritization scheme starts with the most promising features found in the image.

An interesting result from [30] is that the performance of a generic set of 100k visual words obtained from an unrelated dataset is similar to the performance of a specialized vocabulary trained from the descriptors of the points in the corresponding reconstruction. This means that the same vocabulary can be re-used, independently of the considered dataset. The main cause for this somewhat surprising result is that Sattler et al. perform a very approximate nearest neighbor search to compute the assignment of descriptors to visual words to minimize search costs. Specially trained vocabularies do not offer any advantages for such a very approximative search.

Two problems will arise when applying the method from Sattler et al. on even larger datasets. Since multiple SIFT descriptors are stored for every 3D point, the model will eventually become too large to fit into the RAM of a PC. As more and more points are used, the distances between the descriptors of one point and their nearest descriptors belonging to another point decrease. This has a positive impact on the run-time of the RANSAC-based pose estimation, because the SIFT ratio-test is able to remove more and more wrong correspondences. However, as the descriptor space becomes denser, the ratio-test will also filter out more correct correspondences. Thus only features with descriptors very similar to the ones of its corresponding 3D point will pass the SIFT ratio-test. As a result, images differing too much from the views in the database cannot be registered anymore, decreasing the localization performance of the algorithm. Compact models containing fewer points than the original reconstruction require less memory and can therefore help to solve the first problem. Using fewer points can also induce a sparser descriptor space, helping the localization method to avoid rejecting too many good correspondences.

In the case of 3D-to-2D matching, the descriptor space formed by the 2D features in an image is much sparser than the descriptor space of the 3D model. Thus, Li et al. are able to avoid the problem of rejecting too many correct

**Table 1.** Details on the datasets used for experimental evaluation

Dataset	# Cameras	# 3D Points	# Descriptors	# Query Images
Dubrovnik	6044	1,886,884	9,606,317	800
Rome	15, 179	4,067,119	21,515,110	1000
Vienna	1324	1,123,028	4,854,056	266

matches at the cost of finding more wrong correspondences. Note that their approach still has problems scaling to larger datasets. To enable the breadth-first search performed by the algorithm, a larger set of seed points has to be used for reconstructions containing more points. Based to the observation that roughly one out of every 500 points matches by chance, it will happen that the algorithm stops before even considering the whole seed set since enough correspondences are already found. In turn, finding enough good candidate points for matching is not a problem for the method from Sattler et al. due to using a visual vocabulary for finding possible correspondences.

## 4 Compact Models for 2D-to-3D Search

In this section, we evaluate the combination of the localization method from [30] and the point selection scheme proposed by [23]. Specifically, we explore the impact of compact models constructed using different choices for the set cover parameter  $K$  on localization performance, efficiency and accuracy. In Section 4.1 the used datasets and the experimental setup are explained. The impact of the parameter  $K$  on both registration performance and registration times is evaluated in Section 4.2. In Section 4.3 we show that compact models can help the method to handle larger datasets. Section 4.4 details the impact of  $K$  on the localization accuracy. Since the approach from Sattler et al. outperforms the other state-of-the-art approaches, such as [20,23], we do not compare our results against other approaches.

### 4.1 Experimental Setup

We use the three large-scale datasets from [20,23,30] to allow a direct and fair comparison. For two of the datasets, Dubrovnik and Rome, the database images for the reconstruction were obtained from the photo-sharing website Flickr [23]. For the Vienna dataset the database images were taken at regular intervals with a single camera [20]. The original Dubrovnik reconstruction consists of 6844 images depicting parts of the old city of Dubrovnik. 800 randomly selected images were removed from the reconstruction to obtain a set of relevant query images. For every camera in the test set, the SIFT descriptors of the points visible in it were deleted from the model. Any point visible in only one remaining camera was also removed. The query images for the Rome dataset were obtained in the same fashion, removing 1000 randomly selected images from the 16,179 images in the initial reconstruction. In contrast to the Dubrovnik model, the Rome



**Table 2.** The percentage of points selected depending on  $K$  for Dubrovnik and Rome

Dataset	$K$									
	100	200	300	400	500	600	700	800	900	1000
Dubrovnik	3.84%	8.61%	13.58%	18.6%	23.55%	28.24%	32.68%	36.88%	40.82%	44.59%
Rome	3.56%	8.36%	13.57%	18.93%	24.23%	29.42%	34.32%	38.94%	43.29%	47.40%

**Table 3.** The percentage of points selected depending on  $K$  for the Vienna dataset

Dataset	$K$							
	500	750	1000	1250	1500	2000	2500	3000
Vienna	7.54%	12.53%	18.03%	23.62%	29.20%	39.68%	49.28%	58.00%

reconstruction consists of multiple connected components, each one representing a distinct landmark in Rome [23]. The Vienna model consists of 1324 cameras in three connected components. Query images were obtained from the Panoramic website. All query images have maximal width and height of 1600 pixels. The Dubrovnik and Rome models used in [23] and [30] differ slightly in the number of 3D points they contain. We use the latter model. More information about the datasets than presented in Table 1 is available in [20,23].

For the Dubrovnik model, Li et al. computed a transformation into a geo-referenced coordinate frame such that distances in the model can be expressed in meters [23]. Since the query images were obtained by removing images from the initial reconstruction, we can use the original camera positions computed by SfM as ground truth and measure the localization accuracy.

As proposed by Li et al., we accept a query image as localized, or registered against the model, if the best camera pose estimated by RANSAC has at least 12 inliers. Repeating each experiment 10 times to account for the random nature of RANSAC, we report the average number of images that can be registered and the average time needed to register or reject an image. Assuming that SIFT features are already given, the time needed to process an image is the sum of the time needed to assign all of its features to visual words, the time needed for correspondence search and the time needed by RANSAC to estimate the camera pose. Beside the total time, we also report the time required for the correspondence search and the time needed for RANSAC.

## 4.2 Compact Models

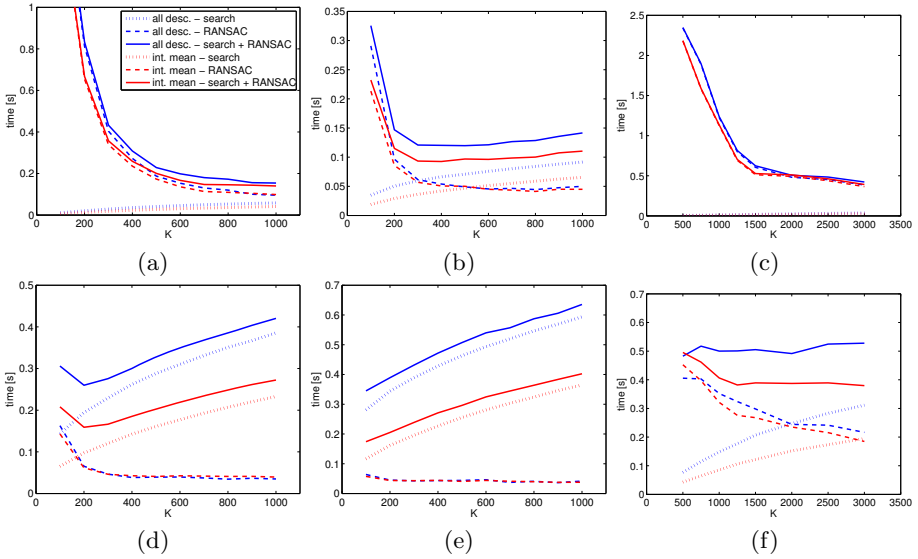
As shown in [38], pictures found on photo collection websites are distributed around certain iconic views as tourists tend to take slightly different pictures of the same buildings. Since the query images for Dubrovnik and Rome were obtained by randomly selecting images from the reconstruction, they have the same distribution as the database images. Thus the descriptors found in the query images should be rather similar to those in the model. While the Vienna model was reconstructed from images taken in nearly regular intervals, the query

**Table 4.** Mean registration performance and times for **100k** visual words and different values for  $K$ .  $K = \infty$  denotes the results reported in [30] for which all available points were used. We obtain a similar registration performance as [30] using compact models with of fewer 3D points. For Dubrovnik and Rome we achieve better registration times.

		all descriptors			integer mean per vw		
		# reg. images	registr. time [s]	rejection time [s]	# reg. images	registr. time [s]	rejection time [s]
Dubrovnik	100	569.40 $\pm$ 3.17	1.79	5.66	604.10 $\pm$ 4.61	1.59	5.45
	200	736.20 $\pm$ 3.26	0.94	5.01	739.60 $\pm$ 1.96	0.78	4.67
	400	776.80 $\pm$ 1.75	0.42	3.43	775.30 $\pm$ 1.16	0.37	3.03
	600	781.30 $\pm$ 1.42	0.31	3.01	778.50 $\pm$ 1.18	0.28	2.66
	800	782.10 $\pm$ 1.20	0.29	2.32	779.20 $\pm$ 1.40	0.26	2.17
	900	782.00 $\pm$ 0.94	0.27	2.45	780.80 $\pm$ 1.23	0.26	1.96
	1000	781.90 $\pm$ 0.99	0.27	2.43	781.30 $\pm$ 0.95	0.25	1.88
	$\infty$	783.90 $\pm$ 1.60	0.31	2.22	782.00 $\pm$ 0.82	0.28	1.70
Rome	100	950.10 $\pm$ 1.66	0.41	3.08	947.40 $\pm$ 2.76	0.32	2.46
	200	965.20 $\pm$ 1.62	0.23	1.84	964.10 $\pm$ 1.45	0.20	1.63
	400	971.90 $\pm$ 1.45	0.21	1.90	972.50 $\pm$ 1.08	0.18	1.77
	600	974.60 $\pm$ 1.07	0.21	1.88	974.70 $\pm$ 1.83	0.18	1.77
	800	973.90 $\pm$ 1.52	0.21	1.76	974.30 $\pm$ 1.16	0.18	1.60
	900	974.00 $\pm$ 1.33	0.22	1.62	975.90 $\pm$ 1.91	0.19	1.56
	1000	974.90 $\pm$ 0.99	0.23	1.63	974.80 $\pm$ 1.87	0.20	1.59
	$\infty$	976.90 $\pm$ 1.29	0.29	1.90	974.60 $\pm$ 1.65	0.25	1.66
Vienna	500	122.50 $\pm$ 2.07	2.44	5.37	127.00 $\pm$ 1.76	2.28	5.12
	1000	181.30 $\pm$ 2.00	1.34	4.25	184.70 $\pm$ 2.54	1.24	4.02
	1500	194.70 $\pm$ 0.82	0.73	3.63	193.90 $\pm$ 1.29	0.64	3.50
	2000	202.30 $\pm$ 1.34	0.62	3.30	202.00 $\pm$ 1.05	0.63	3.04
	2500	206.40 $\pm$ 1.26	0.60	3.07	205.10 $\pm$ 1.10	0.58	2.85
	3000	206.90 $\pm$ 0.74	0.54	2.84	206.10 $\pm$ 1.10	0.51	2.70
	$\infty$	207.70 $\pm$ 1.06	0.50	2.40	206.90 $\pm$ 0.88	0.46	2.43

images obtained from Panoramio follow a different distribution. Furthermore, the database images were taken with a single camera on the same day while query images are taken at different days and at different times of day with different cameras. This makes the Vienna dataset the most challenging of the three datasets and we can expect a larger difference between the SIFT descriptors found in the query image and the those found in the database images. Due to this difference in distributions, we use a different range of values for the set cover parameter  $K$  for the Vienna dataset compared to the Dubrovnik and Rome datasets, similar to [23]. Table 2 shows the percentage of points selected depending on  $K$  for the Dubrovnik and Rome datasets, while Table 3 shows the percentage of selected points for the Vienna dataset. We only consider values for  $K$  until obtaining around 50% of the points contained in the original model since we want to use the compact models to save storage space.

We evaluate the compact models obtained for the values for  $K$  shown in Tables 2 and 3 together with the two strategies, *all desc.* and *int. mean*, in the



**Fig. 1.** Dependence of the average time needed to find enough correspondences and the average time to compute the camera pose through RANSAC on the parameter  $K$ . Timings are shown for (a), (d) Dubrovnik, (b), (e) Rome and (c), (f) Vienna. 100k visual words were used for the results shown in the top row and 10k words for the bottom row. Fewer 3D points yield more wrong correspondences, increasing the run-time of RANSAC. Search time increases with the number of points in the words.

pipeline proposed in [30]. The visual vocabulary containing 100k words employed in this experiments is the same as in [30]. We report the mean number of images that can be localized and the mean time needed to register or reject an image in Table 4. Small values for  $K$  lead to a significantly worse localization performance with high registration and rejection times. Using more points allows to achieve a registration performance similar to [30]. Slightly faster registration times can be achieved for  $K$  from {800, 900, 1000} for Dubrovnik and Rome, while the registration times for the Vienna dataset are a little worse compared to the full model. There are two possible explanations for the observed behavior. First, using fewer points and thus fewer descriptors leads to a sparser descriptor space and visual words that are less full. As the distances between descriptors stored in a visual word grow, it becomes more likely to accept wrong matches through the SIFT ratio-test, which in turn increase the registration time. Secondly, the selected points might not suffice to allow robust localization.

The first explanation is easy to verify. Figure 1 shows how the mean time for correspondence search and the mean time RANSAC needs depend on the choice of  $K$  for (a) Dubrovnik, (b) Rome and (c) Vienna. Compact models with more points indeed lead to faster RANSAC times due to fewer wrong matches.

To reject the second explanation, we repeat the experiment using a visual vocabulary containing only 10k words. Since each of the words in this smaller vocabulary covers a larger part of descriptor space, the likelihood of assigning

**Table 5.** Mean registration performance and times for **10k** visual words and different values for  $K$ . For the Dubrovnik and Rome datasets, fewer points still allow a similar registration performance compared to [30] at higher localization costs. A significantly better performance at comparable registration times is achieved for the Vienna dataset.

		all descriptors			integer mean per vw		
		# reg. images	registr. time [s]	rejection time [s]	# reg. images	registr. time [s]	rejection time [s]
Dubrovnik	$K$						
	100	771.00 $\pm$ 1.49	0.36	2.03	765.40 $\pm$ 1.96	0.26	1.55
	200	778.80 $\pm$ 1.75	0.31	1.81	777.20 $\pm$ 0.92	0.21	1.18
	300	780.70 $\pm$ 0.95	0.33	1.70	778.00 $\pm$ 1.49	0.22	1.18
	400	782.60 $\pm$ 1.84	0.35	1.86	779.20 $\pm$ 1.03	0.24	1.19
	600	783.30 $\pm$ 0.95	0.40	2.02	781.60 $\pm$ 2.22	0.27	1.38
	800	783.40 $\pm$ 0.97	0.44	2.12	783.20 $\pm$ 1.62	0.30	1.43
	1000	784.50 $\pm$ 1.65	0.47	2.22	784.30 $\pm$ 0.82	0.32	1.69
	[30]	783.90 $\pm$ 1.60	0.31	2.22	782.00 $\pm$ 0.82	0.28	1.70
Rome	100	964.20 $\pm$ 1.32	0.38	1.54	959.00 $\pm$ 1.56	0.21	1.13
	200	972.40 $\pm$ 1.84	0.43	2.01	968.90 $\pm$ 1.10	0.24	1.36
	300	974.90 $\pm$ 0.99	0.47	2.38	971.60 $\pm$ 1.17	0.28	1.44
	400	978.80 $\pm$ 1.03	0.51	2.64	974.80 $\pm$ 1.55	0.31	1.55
	600	980.00 $\pm$ 0.67	0.58	2.76	976.60 $\pm$ 0.84	0.36	1.86
	800	978.80 $\pm$ 1.03	0.63	3.19	976.80 $\pm$ 1.62	0.40	2.09
	1000	980.20 $\pm$ 1.75	0.67	3.45	977.10 $\pm$ 1.73	0.44	2.23
		[30]	976.90 $\pm$ 1.29	0.29	1.90	974.60 $\pm$ 1.65	0.25
Vienna	500	198.70 $\pm$ 1.16	0.54	2.63	198.10 $\pm$ 1.66	0.55	2.18
	750	209.00 $\pm$ 1.25	0.57	2.31	205.90 $\pm$ 0.74	0.52	1.86
	1000	215.00 $\pm$ 1.25	0.55	1.91	209.60 $\pm$ 1.43	0.46	1.61
	1250	217.50 $\pm$ 0.97	0.56	1.67	213.60 $\pm$ 0.70	0.44	1.37
	1500	218.10 $\pm$ 0.88	0.56	1.54	213.90 $\pm$ 1.10	0.44	1.35
	2000	219.30 $\pm$ 0.67	0.55	1.43	214.80 $\pm$ 1.23	0.44	1.27
	2500	219.70 $\pm$ 0.82	0.58	1.31	215.20 $\pm$ 0.92	0.44	1.23
	3000	218.90 $\pm$ 1.20	0.58	1.29	214.10 $\pm$ 1.29	0.43	1.23
	[30]	207.70 $\pm$ 1.06	0.50	2.40	206.90 $\pm$ 0.88	0.46	2.43

the descriptors of a 3D point and the descriptor of its corresponding feature to the same word increases compared to the original vocabulary. Table 5 shows that a good registration performance can be achieved for much lower values of  $K$  with the smaller vocabulary, indicating that enough points for robust localization are selected. A compact model containing only about 18% of the original points ( $K = 400$  for Dubrovnik and Rome,  $K = 1000$  for Vienna) gives a performance comparable to the original methods from [30], albeit at increased registration times. As shown in Figure 1(d)-(e) this increase is mainly due to the slower search as more points are contained in the words. It is noticeable that the difference in search time for both strategies is much larger for 10k words than for 100k words. Since more descriptors of the same point are mapped to the same word for the smaller vocabulary, the *integer mean* strategy is able to compress them into one mean descriptor while the *all descriptor* strategy has to use all descriptors. At the

**Table 6.** Results for combining different versions of the three models for query images from each dataset. We use  $K_1$  to build compact models for Dubrovnik and Rome and  $K_2$  to obtain a compact model for Vienna. For comparison we include the results from [30] on the single models. Due to the denser descriptor space, registration performance drops compared to [30] for the combined models, but the usage of compact models can help to decrease the registration and rejection times at a similar localization performance.

$K_1 / K_2$		method	# reg. images	registered			rejection time [s]
				search [s]	RANSAC [s]	total [s]	
Dubrovnik	$\infty / \infty$	all desc.	779.20 $\pm$ 0.63	0.42	0.02	0.55	1.32
		int. mean	776.00 $\pm$ 1.25	0.33	0.02	0.46	1.05
	900 / 2500	all desc.	775.80 $\pm$ 1.23	0.27	0.02	0.40	0.90
		int. mean	775.80 $\pm$ 1.40	0.20	0.02	0.33	0.68
	400 / 1000	all desc.	774.60 $\pm$ 1.17	0.19	0.02	0.31	0.64
		int. mean	773.50 $\pm$ 0.85	0.13	0.02	0.25	0.45
	[30]	all desc.	783.90 $\pm$ 1.60	0.10	0.08	0.31	2.22
		int. mean	782.00 $\pm$ 0.82	0.08	0.08	0.28	1.70
Rome	$\infty / \infty$	all desc.	973.10 $\pm$ 2.02	0.24	0.04	0.36	1.68
		int. mean	971.20 $\pm$ 1.55	0.19	0.04	0.31	1.35
	900 / 2500	all desc.	975.00 $\pm$ 1.25	0.16	0.04	0.28	1.32
		int. mean	970.20 $\pm$ 1.23	0.12	0.04	0.24	1.10
	400 / 1000	all desc.	971.90 $\pm$ 0.74	0.11	0.04	0.23	1.26
		int. mean	970.90 $\pm$ 1.79	0.07	0.04	0.20	1.09
	[30]	all desc.	976.90 $\pm$ 1.29	0.15	0.05	0.29	1.90
		int. mean	974.60 $\pm$ 1.65	0.11	0.05	0.25	1.66
Vienna	$\infty / \infty$	all desc.	202.70 $\pm$ 0.67	0.54	0.01	0.67	1.29
		int. mean	200.80 $\pm$ 0.79	0.43	0.01	0.57	0.98
	900 / 2500	all desc.	203.90 $\pm$ 0.74	0.36	0.02	0.50	0.82
		int. mean	200.60 $\pm$ 0.52	0.27	0.03	0.41	0.60
	400 / 1000	all desc.	192.60 $\pm$ 1.26	0.24	0.02	0.37	0.66
		int. mean	189.10 $\pm$ 0.57	0.16	0.02	0.30	0.48
	[30]	all desc.	207.70 $\pm$ 1.06	0.06	0.30	0.50	2.40
		int. mean	206.90 $\pm$ 0.88	0.05	0.28	0.46	2.43

same time, the *all descriptor* strategy is able to handle denser visual words much better as all information about the 3D points is preserved, which is visible in the better registration performance for smaller values for  $K$ . We observe a significant increase in the localization performance for the smaller vocabulary on the Vienna dataset. As mentioned above, the difference in viewpoint and viewing condition is the largest on this dataset, explaining that using fewer words increases the chance of assigning features and points that belong together to the same visual word. As predicted, the number of wrong correspondences decreases for the words in the smaller vocabulary as evident by the faster RANSAC run-time shown in Figure 1(d)-(f) compared to Figure 1(a)-(c). This faster pose estimation has the largest impact on the Vienna dataset for which the mean registration time was dominated by the time spend by RANSAC when using 100k words.

### 4.3 Combining the Datasets

As shown in the previous experiments, we can use compact representations of the 3D models obtained by the point selection scheme from [23] to reduce the memory footprint and still obtain a similar registration performance and efficiency compared to the original models. For larger datasets, the descriptor space becomes denser as more points are used. As a result, the SIFT ratio-test is more likely to also reject good correspondences. As compact models contain fewer points, they could help to avoid the loss in registration performance.

In this section we want to explore the effect of using compact models on the density of the descriptor space for datasets larger than the three models used so far. Although modern SfM approaches can efficiently handle large datasets, obtaining the images for very large scenes is still challenging. We therefore try to simulate a larger dataset by combining the three models. This is motivated by the observation that only few correspondences are found between points in one model and query images from another dataset [30]. The combined datasets therefore represents a sort of "best case" model which consists of distinct landmarks. If we can observe that the descriptor space becomes too dense for this model, we would expect that the space will also become too dense for other large datasets.

We combine (subsets of) the three models to obtain three larger datasets: The first one consists of all points from all three datasets, i.e., we set  $K = \infty$ . The second is obtained using the point selection scheme with  $K = 900$  on Dubrovnik and Rome and  $K = 2500$  on Vienna. The last one consists of the points selected with  $K = 400$  on Dubrovnik and Rome and  $K = 1000$  on Vienna. We chose the combinations 900 / 2500 and 400 / 1000 because these were the smallest values for  $K$  that gave results similar to the original method when using 100k respectively 10k visual words. We only consider the vocabulary of size 100k words since the search time for 10k words were already too large for the single models. Table 6 reports the registration performance and efficiency for the query images from each dataset and compares it to the results obtained in [30] on the single models. As can be seen, the sparser descriptor space obtained from the compact models is still too dense to prevent a loss in registration performance. However, the compact models can be used to speed up the search times while still obtaining very similar performance compared to using the full model.

The denser descriptor space has a significant impact on the pose estimation time as most wrong correspondences are eliminated by the SIFT ratio-test, allowing us to achieve even better registration and rejection times than the original method. For example, we obtain significantly better registration times with  $K = 1000$  for the query images from Vienna dataset when combining the models compared to only considering the Vienna model.

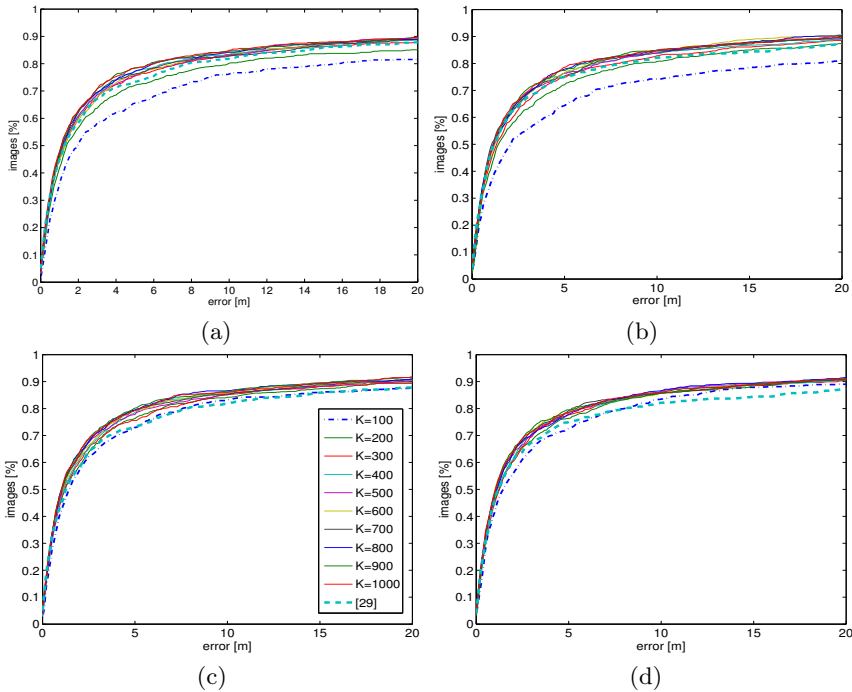
### 4.4 Localization Accuracy

We measure the localization accuracy of the combination of point selection and the localization method from [30] on the Dubrovnik dataset. The random nature of RANSAC results in slightly differing camera pose estimates for all repetitions

of the experiment. To compensate for this, we measure the average camera position for every query image from all its estimated poses with at least 12 inliers from the 10 repetitions. We report the distance between this averaged position and the ground truth position of the query camera in the original reconstruction.

The greedy point selection algorithm iteratively picks the point that covers the largest number of cameras that have not yet been covered and thus prefers points visible in many cameras [23]. We can expect that the amount of positional uncertainty related to the selected points is relative small, since they have been detected in multiple images. Using these high-quality points should improve the localization accuracy. Unfortunately, SIFT features are not equally distributed over images but mostly found in highly textured regions. Given such a highly textured region, it is rather likely that multiple points in this region appear in many database images. Thus if one of them is selected by the greedy algorithm it is very likely that also the other points are selected since they are also visible in a similar number of images. As a result, it might happen that the selected points are not well-distributed over the model but form small clusters. This in turn can lead to unstable or even degenerate configurations for the pose estimation step. To verify whether using fewer points yields less accurate localization results, we look at the cumulative distribution of the query images over localization errors depicted in Figure 2. In contrast to [30], we followed RANSAC-based pose estimation with a linear least-squares estimate of the pose from the inliers. As seen in Section 4.2, the number of images that can be registered differs with the choice of  $K$ . To allow a fair comparison, we normalized the cumulative histogram for each variant using the total number of images that it could register, i.e., the number of images that could be localized at least once during the 10 repetitions of the experiment. As can be seen in the figure, using too few points indeed results in worse localization accuracy. However, about 14% of the total features ( $K = 300$ , cf. Table 2) are already sufficient to achieve localization accuracy comparable to or better than the results reported in [30]. Choosing  $K$  from  $\{800, 900, 1000\}$  gives the best results. We notice that using the smaller vocabulary of 10k words improves the accuracy. Due to the coarser quantization and the approximative nature of visual word assignments, it is more likely to assign two descriptors of the same 3D point to the same visual word when using 10k words instead of 100k. This enables the algorithm to find more correspondences for points seen from rather large viewpoint changes compared to the original cameras, which in turn yield better configurations for pose estimation.

More details on selected values for  $K$  are given in Table 7. We report the median localization error, the 1st and 3rd quartile and the number of images with a localization error smaller than 18.3m respectively 400m, which correspond to the mean and maximal errors reported in [23]. The results verify the observations from Figure 2, since compact models help to improve the localization accuracy. Again, the usage of a smaller vocabulary has a positive impact on the accuracy of the position estimates. We do not report the mean or maximal registration error, since there are a few images with very high localization error of up to multiple kilometers. These large errors are caused by degenerate point configurations for



**Fig. 2.** Normalized cumulative histograms of the distribution of the localization error depending on  $K$  for *all descriptors* using (a) 100k words respectively (c) 10k words and *integer mean* using (b) 100k words and (d) 10k words. Choosing  $K \geq 300$  helps to improve the localization accuracy compared to the original method independently of the vocabulary size since a higher percentage of reliably localized images points is used. Values for  $K$  from the range [800, 1000] give the best results.

pose estimation. We observe that images with such large errors mostly have more than 12 inliers, indicating that the pure inlier count is not a good measure for localization accuracy. This behavior has already been reported by Sattler et al. [30]. They show that using the focal length of an image, obtained from its EXIF tag, for 3-point pose estimation [13, 16] or a more restrictive camera model, which estimates only its focal length and a radial distortion parameter [21], help to obtain more accurate estimates. We could also estimate the covariance of the position parameters of the query camera and reject a camera for which the positional uncertainty is too high.

We report the localization accuracies for the combined datasets in Table 8. The results were obtained without the final linear least-square pose estimate and show no significant difference in localization accuracy between the different combinations and the original results from [30], obtained using only the Dubrovnik dataset. The drop in localization accuracy compared to Table 7 can be explained by the different set of correspondences found when also using the points from the other datasets.



**Table 7.** Statistics on the localization errors for selected values of  $K$ . Using compact models helps to improve the localization accuracy compared to the original methods using all points ( $K = \infty$ ) from [30] and the method from [23].

$K$	Method	# vw	Median	Quartiles [m]		#imgs. with error	
			[m]	1st	3rd	< 18.3m	> 400m
400	all desc.	10k	1.2	0.5	4.1	710	7
		100k	1.3	0.5	4.3	690	9
	int. mean	10k	1.2	0.5	4.1	703	6
		100k	1.3	0.5	4.5	689	12
800	all desc.	10k	1.1	0.4	3.8	710	9
		100k	1.2	0.5	4.3	698	11
	int. mean	10k	1.2	0.4	4.3	714	12
		100k	1.3	0.4	4.1	705	13
900	all desc.	10k	1.1	0.4	3.6	713	8
		100k	1.2	0.4	3.9	698	10
	int. mean	10k	1.2	0.5	3.5	709	9
		100k	1.3	0.5	4.3	696	14
1000	all desc.	10k	1.1	0.4	3.8	714	9
		100k	1.2	0.4	4.0	700	11
	int. mean	10k	1.1	0.4	4.1	711	11
		100k	1.3	0.5	4.3	701	10
$\infty$	all desc.	100k	1.4	0.4	5.9	685	16
	int. mean	100k	1.3	0.5	5.1	675	13
100	P2F [23]	-	9.3	7.5	13.4	655	-

**Table 8.** Statistics on the localization errors for the combined datasets from Section 4.3. There is no significant difference in localization accuracy between the different combinations and the original results from [30].

$K_1 / K_2$	Method	Median	Quartiles [m]		#imgs. with error	
		[m]	1st	3rd	< 18.3m	> 400m
$\infty / \infty$	all desc.	1.4	0.5	4.7	688	13
	int. mean	1.3	0.4	5.2	674	9
900 / 2500	all desc.	1.3	0.4	5.8	671	12
	int. mean	1.5	0.5	5.5	677	11
400 / 1000	all desc.	1.5	0.5	6.4	671	12
	int. mean	1.5	0.5	6.9	671	13
[30]	all desc.	1.4	0.4	5.9	685	16
	int. mean	1.3	0.5	5.1	675	13

## 5 Conclusion and Future Work

In this paper we have shown that not all points contained in a Structure-from-Motion model are needed for robust image-based localization. By combining the state-of-the-art localization method from Sattler et al. [30] and the simple point

selection scheme from Li et al. [23] we demonstrated that using less than half of the original points still allows state-of-the-art localization performance at similar registration and rejection times and with slightly better localization accuracy. This result is still valid when combining the different datasets to simulate one larger reconstruction. Therefore, we can save memory by storing fewer points and descriptors without a significant sacrifice in performance and efficiency. As the method of computing the compact models does not depend on the type of feature descriptor, it can be readily combined with more memory efficient descriptors [5,36] to further reduce the memory footprint. The point selection algorithm from Li et al. might prefer points that form small clusters over points that are well-distributed over the model, which can lead to unstable configurations for pose estimation. The point selection scheme does not take similarity in descriptor appearance into account. As shown in Section 4.3, it thus cannot prevent a drop in registration performance when the descriptor space becomes denser. Furthermore, localization performance and efficiency depend on the set cover parameter  $K$ . An interesting open question is whether we can design a better, parameter-free point filtering algorithm that ensures a better distribution of points and impacts the descriptor space.

As shown in Section 4.2, the number of points stored in a visual word has an impact on the quality of the found correspondences. A data structure that tries to adapt the number of words to take the density of the points inside a word into account could help to improve localization performance further.

Finally, we notice that the localization methods proposed by Li et al. and Sattler et al. have both distinct strength and weaknesses, as detailed at the end of Section 3. Combining their matching directions could help to obtain a novel localization method that combines the strength of both approaches while eliminating their weaknesses, which would have a positive impact on its performance.

**Acknowledgments.** We gratefully acknowledge support by UMIC (DFG EXC 89) and Mobile ACcess (EFRE 280401102). We thank all participants of the Dagstuhl Seminar 11261, "Outdoor and Large-Scale Real-World Scene Analysis", for their encouraging comments and helpful and interesting discussions.

## References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a Day. In: IEEE 12th International Conference on Computer Vision, pp. 72–79. IEEE (2009)
2. Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D.: Wide Area Localization on Mobile Phones. In: 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 73–82. IEEE Comp. Society, Washington, DC (2009)
3. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. *J. ACM* 45, 891–923 (1998)

4. Avrithis, Y., Kalantidis, Y., Tolias, G., Spyrou, E.: Retrieving Landmark and Non-Landmark Images from Community Photo Collections. In: Proceedings of the International Conference on Multimedia, pp. 153–161. ACM, New York (2010)
5. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding* 110, 346–359 (2008)
6. Castle, R.O., Klein, G., Murray, D.W.: Video-rate Localization in Multiple Maps for Wearable Augmented Reality. In: 12th IEEE International Symposium on Wearable Computers, pp. 15–22 (2008)
7. Chen, D.M., Baatz, G., Köser, K., Tsai, S.S., Vedantham, R., Pylvänäinen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., Girod, B., Grzeszczuk, R.: City-scale Landmark Identification on Mobile Devices. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 737–744. IEEE (2011)
8. Chum, O., Matas, J., Obdržálek, S.: Enhancing RANSAC by Generalized Model Optimization. In: Hong, K.-S., Zhang, Z. (eds.) Proceedings of the Asian Conference on Computer Vision, vol. 2, pp. 812–817. Asian Fed. of Comp. Vis. Societies (2004)
9. Chum, O., Matas, J.: Optimal Randomized RANSAC. *Trans. Pattern Analysis and Machine Intelligence* 30, 1472–1482 (2008)
10. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.P.: Discrete-Continuous Optimization for Large-Scale Structure from Motion. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3001–3008. IEEE (2011)
11. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. Robotics Research* 27, 647–665 (2008)
12. Eade, E., Drummond, T.: Scalable Monocular SLAM. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 469–476. IEEE Comp. Society, Washington, DC (2006)
13. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. ACM* 24, 381–395 (1981)
14. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010)
15. Gammeter, S., Bossard, L., Quack, T., Van Gool, L.: I know what you did last summer: object-level auto-annotation of holiday snaps. In: IEEE 12th International Conference on Computer Vision, pp. 614–621. IEEE (2009)
16. Haralick, R.M., Lee, C.-N., Ottenberg, K., Nölle, M.: Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *Int. J. Comp. Vision* 13, 331–356 (1994)
17. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
18. Havlena, M., Torii, A., Pajdla, T.: Efficient Structure from Motion by Graph Optimization. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 100–113. Springer, Heidelberg (2010)
19. Hays, J., Efros, A.A.: IM2GPS: estimating geographic information from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)

20. Irschara, A., Zach, C., Frahm, J.-M., Bischof, H.: From Structure-from-Motion Point Clouds to Fast Location Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2599–2606. IEEE (2009)
21. Josephson, K., Byröd, M.: Pose Estimation with Radial Distortion and Unknown Focal Length. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2419–2426. IEEE (2009)
22. Knopp, J., Sivic, J., Pajdla, T.: Avoiding Confusing Features in Place Recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 748–761. Springer, Heidelberg (2010)
23. Li, Y., Snavely, N., Huttenlocher, D.P.: Location Recognition Using Prioritized Feature Matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 791–804. Springer, Heidelberg (2010)
24. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comp. Vision* 60, 91–110 (2004)
25. Muja, M., Lowe, D.G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: International Conference on Computer Vision Theory and Application, pp. 331–340. INSTICC Press (2009)
26. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2161–2168. IEEE (2006)
27. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
28. Pollefeys, M., Nister, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.-J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed Real-Time Urban 3D Reconstruction From Video. *Int. J. Comp. Vision* 78, 143–167 (2008)
29. Robertson, D., Cipolla, R.: An Image-Based System for Urban Navigation. In: Hoppe, A., Barman, S., Ellis, T. (eds.) The 15th British Machine Vision Conference, pp. 819–828. BMVA (2004)
30. Sattler, T., Leibe, B., Kobbelt, L.: Fast Image-Based Localization using Direct 2D-to-3D Matching. In: IEEE 13th International Conference on Computer Vision, pp. 667–674. IEEE (2011)
31. Schindler, G., Brown, M., Szeliski, R.: City-Scale Location Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7. IEEE (2007)
32. Stephen, S., Lowe, D.G., Little, J.: Global Localization using Distinctive Visual Features. In: International Conference on Intelligent Robots and Systems, pp. 226–231 (2002)
33. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 1470–1477. IEEE Comp. Society, Washington, DC (2003)
34. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. In: SIGGRAPH Conference Proceedings, pp. 835–846. ACM, New York (2006)
35. Strecha, C., Pylvanainen, T., Fua, P.: Dynamic and Scalable Large Scale Image Reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 406–413. IEEE (2010)
36. Strecha, C., Bronstein, A.M., Bronstein, M.M., Fua, P.: LDAHash: Improved matching with smaller descriptors. EPFL-REPORT-152487 (2010)

37. Wendel, A., Irschara, A., Bischof, H.: Natural Landmark-based Monocular Localization for MAVs. In: IEEE International Conference on Robotics and Automation, pp. 5792–5799. IEEE (2011)
38. Weyand, T., Leibe, B.: Discovering Favorite Views of Popular Places with Iconoid Shift. In: IEEE 13th International Conference on Computer Vision, pp. 1132–1139. IEEE (2011)
39. Zamir, A.R., Shah, M.: Accurate Image Localization Based on Google Maps Street View. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 255–268. Springer, Heidelberg (2010)
40. Zhang, W., Kosecka, J.: Image Based Localization in Urban Environments. In: 3rd International Symposium on 3D Data Processing, Visualization and Transmission, pp. 33–40. IEEE Comp. Society, Washington, DC (2006)

# Perspective and Non-perspective Camera Models in Underwater Imaging – Overview and Error Analysis

Anne Sedlazeck\* and Reinhard Koch

Multimedia Information Processing, Institute of Computer Science,  
Christian-Albrechts-University (CAU) of Kiel, Germany

{sedlazeck,rk}@mip.informatik.uni-kiel.de

<http://www.mip.informatik.uni-kiel.de>

**Abstract.** When capturing images underwater, image formation is affected in two major ways. First, the light rays traveling underwater are absorbed and scattered depending on their wavelength, creating effects on the image colors. Secondly, the glass interface between air and water refracts the ray entering the camera housing because of a different index of refraction of water, hence the ray is also affected in a geometrical way.

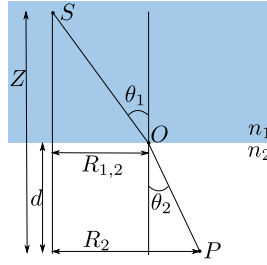
This paper examines different camera models and their capabilities to deal with geometrical effects caused by refraction. Using imprecise camera models leads to systematic errors when computing 3D reconstructions or otherwise exploiting geometrical properties of images. In the literature, many authors have published work on underwater imaging by using the perspective pinhole camera model (single viewpoint model - SVP) with a different effective focal length and distortion to compensate for the error induced by refraction at the camera housing. On the other hand, methods were proposed, where refraction is modeled explicitly or where generic, non-single-view-point camera models are used. In addition to discussing all three model categories, an accuracy analysis of using the perspective model on underwater images is given and shows that the perspective model leads to systematic errors that compromise measurement accuracy.

## 1 Introduction

Underwater imaging is becoming more and more popular as technology becomes available to research the ocean floor at great water depths. Exemplary applications are the measurement of fish sizes or other organisms - in general observations of different ecosystems, (volumetric) measurements of deep sea structures like hydrothermal vents, offshore oil production, construction and maintenance of offshore wind parks, cable and pipe inspection, underwater archeology (e.g. ship wreck inspection, cave diving), and ship hull inspection as a measure against terrorism.

---

\* This work was supported by the German Research Foundation (DFG), KO-2044/6-1 3D Modeling of Seafloor Structures from ROV-based Video Sequences.



**Fig. 1.** Fermat's principle based on the ray from  $S$  to  $P$  being refracted at  $O$

In contrast to conventional computer vision, underwater image formation is effected in two ways. First, while traveling through the water, the light rays are partly absorbed and scattered, dependent on the wavelength. This leads to a green or blue hue on underwater images and has thus an effect on the colors. Secondly, refraction of light occurs at the boundary to the underwater housing, since the inside is usually occupied by air. Refraction effects the geometry of the image formation and is the subject of this work.

### 1.1 Refraction at Underwater Housings

The definition of refraction, as in [20], is the deviation of a light ray from its former path when entering a medium with a new optical density. While the frequency is constant, this causes the propagation velocity to change and all rays, not traveling perpendicularly to the interface, change their direction and enter the new medium under a different angle compared to the interface's normal. This effect is explained by Fermat's principle: the light traveling through two different media always travels the way that takes the least time to traverse. A derivation using the distances traveled and the speed of light in the different media yields Snell's law.

Following figure 1, the time the ray needs to travel from  $S$  to  $P$  is described by the following sum:

$$t = \frac{\sqrt{(Z-d)^2 + R_{1,2}^2}}{\nu_1} + \frac{\sqrt{d^2 + (R_2 - R_{1,2})^2}}{\nu_2}, \quad (1)$$

where  $\nu_1$  and  $\nu_2$  denote the speed of light in the corresponding medium. In order to minimize this equation, its derivative is computed:

$$\frac{\partial t}{\partial R_{1,2}} = \frac{R_{1,2}}{\nu_1 \sqrt{(Z-d)^2 + R_{1,2}^2}} + \frac{-(R_2 - R_{1,2})}{\nu_2 \sqrt{d^2 + (R_2 - R_{1,2})^2}} = 0, \quad (2)$$

which can also be expressed by:

$$\frac{\sin \theta_1}{\nu_1} = \frac{\sin \theta_2}{\nu_2}, \quad (3)$$

**Table 1.** Indexes of refraction for air, different kinds of water, and glass as in [20] p. 163 and [36] p. 85

Medium	Index of Refraction
air ( $\lambda = 589nm$ )	1.0003
pure water ( $\lambda = 700nm, 30^\circ C, p = 1.01e10^9 Pa$ )	1.329
pure water ( $\lambda = 700nm, 30^\circ C, p = 1.08e10^8 Pa$ )	1.343
sea water ( $\lambda = 700nm, 30^\circ C, p = 1.01e10^9 Pa$ )	1.335
sea water ( $\lambda = 400nm, 30^\circ C, p = 1.08e10^8 Pa$ )	1.363
quartz glass ( $\lambda = 589nm$ )	1.4584
acrylic glass (Plexiglas, $\lambda = 589nm$ )	1.51
crown glass ( $\lambda = 589nm$ )	1.52
light flint glass ( $\lambda = 589nm$ )	1.58
dense flint glass ( $\lambda = 589nm$ )	1.66
Lanthan flint glass ( $\lambda = 589nm$ )	1.80

and with  $c$  being the speed of light in vacuum and  $n_1 = c/v_1$  and  $n_2 = c/v_2$ , Snell's law follows:

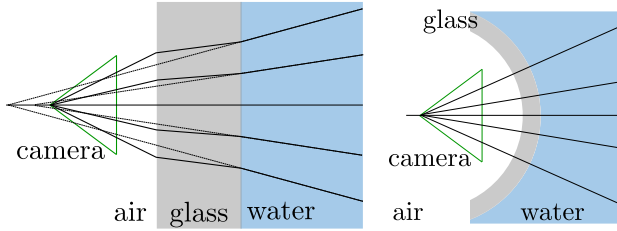
$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{n_2}{n_1}. \quad (4)$$

$n_1$  and  $n_2$  are called indexes of refraction describing the phenomenon for both media. When setting the index of refraction to 1.0 for vacuum, all other indexes are determined relative to it. Important for this work are the indexes of refraction for water, glass, and air. The index of air is close to the index of vacuum, and is therefore usually set to 1.0. The index of water changes due to wavelength, salinity, pressure, and temperature, causing slight changes of the index of refraction when comparing different water bodies in the ocean (see table [1]). According to [36], the dependency on all four parameters only induces a change of about 3% in the index of refraction in the whole relevant parameter range for ocean optics, thus the change can be ignored. In contrast to that, [20] lists the indexes of refraction for glass (see table [1]) and shows a far stronger variation depending on the different materials, usually requiring them to be considered explicitly.

When using cameras to capture underwater images, those cameras need to be put into watertight underwater housings. These underwater housings have a piece of glass through which the image is taken, while the inside of the housing is filled with air. Hence, refraction, as described above, happens twice: first, at the water-glass interface and, secondly, at the glass-air interface (fig. 2 left), causing the ray to shift due to the double refraction depending on the glass thickness.

When working with camera housings, two different kinds of glass ports need to be considered. Planar glass, effecting most of the rays to be refracted just as depicted in figure 2 on the left, and dome ports (fig. 2 right), eliminating the refractive effect to some extent. In theory, the dome port completely removes refraction, due to zero angles between the interface normal and incoming rays.





**Fig. 2.** Left: refraction at flat glass interface. Right: straight rays entering the underwater housing through a dome port.

However, the port and housing need to be manufactured and assembled such that the camera is centered perfectly with respect to the dome port's center for this to work. In case of a flat port or an imperfectly fit dome port, refraction of light rays invalidates the single-view-point camera model. This can be observed in figure 2 on the left for a flat port: the rays traveling in the water in the camera's direction are traced towards the optical center without refraction (dashed lines) and they do not intersect the optical axis in one common center of projection. Hence, the camera does not have a single view point (non-SVP camera model) and the commonly used pinhole camera model is invalid for underwater images.

In the literature, a large group of authors uses the perspective model, although their camera housings have flat ports, while others seek a complete physical model of the refraction effects to achieve greater accuracy. A third approach consists of using a more generic camera model, not requiring a single view point, only being based on rays. The goal of this work is to examine the wealth of approaches to underwater imaging and to discuss their benefits and shortcomings. We will show that the SVP assumption is not sufficient and will discuss a camera model that eliminates these shortcomings.

Sections 2-4 will analyze in depth the state of the art in underwater camera models and will give an overview of the publications on the above mentioned categories. A concise summary of all papers and their application area is given in tables 5-7 in an overview covering perspective models (28 papers), ray-based models (6 papers), and physical, refractive models (19 papers). In section 5, an error analysis of the usage of the perspective and the physical imaging model on underwater images is presented, followed by a conclusion.

## 2 The Perspective Camera Model

Throughout the article, geometric entities are described in a common notation, summarized in table 2. In addition, the conversion of Euclidean coordinates into cylinder coordinates is required:

$$\begin{pmatrix} R \\ \varphi \\ Z \end{pmatrix} = \begin{pmatrix} \sqrt{X^2 + Y^2} \\ \arccos\left(\frac{X}{R}\right) \\ Z \end{pmatrix}. \quad (5)$$

**Table 2.** Notations for rays and points in Euclidean, homogeneous, and cylinder coordinates. Note that in some cases, it is sufficient to use the radial coordinates  $(R, Z)^T$ , thus  $\varphi$  is omitted for the sake of readability.

Homogeneous Point in 3D	$\mathbf{X} = (X, Y, Z, 1)^T$
Homogeneous Point in 2D	$\mathbf{x} = (x, y, 1)^T$
Euclidean Vector in 3D	$\mathbf{X} = (X, Y, Z)^T$
Euclidean Vector in 2D	$\mathbf{x} = (x, y)^T$
Ray in 3D	$\check{\mathbf{X}} = (\check{X}, \check{Y}, \check{Z})^T$
3D vector in cylinder coordinates	$\mathbf{X}^c = (R, \varphi, Z)^T$
Ray in cylinder coordinates	$\check{\mathbf{X}}^c = (\check{R}, \check{\varphi}, \check{Z})^T$
distance camera center - interface in <i>mm</i>	$d$
glass thickness in <i>mm</i>	$d_g$
indexes of refraction (air, glass, water)	$n_a, n_g, n_w$

The pinhole camera model with distortion is one of the established models for perspective cameras. It uses rays to describe how 3D points are projected to individual pixels and is parametrized by intrinsic parameters describing the camera's internal properties:

$$\mathbf{K} = \begin{pmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

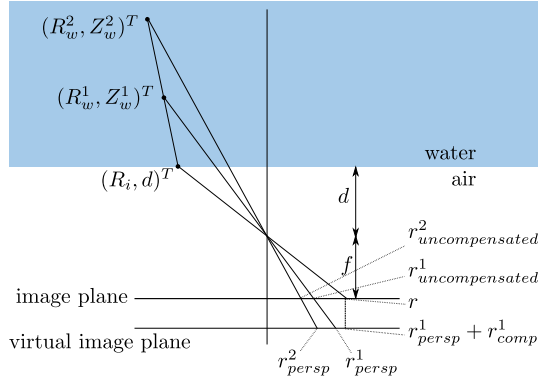
with  $f$  being the focal length,  $a$  being the aspect ratio,  $s$  being the skew, and  $(c_x, c_y)$  being the principal point. Extrinsic parameters describe the camera pose, thus, the projection matrix follows with  $\mathbf{R}$  being an orthonormal rotation matrix and  $\mathbf{C}$  being a translation vector:  $P = \mathbf{KR}^T[\mathbf{I} - \mathbf{C}]$ . A homogeneous point  $\mathbf{X}$  in 3D space is projected by the camera, resulting in a homogeneous 2D point  $\mathbf{x} = \mathbf{PX}$ . In addition, it is possible to use this parametrization to back project 2D image points, i.e. to compute the ray in space on which the 3D point lies [18]. Imperfect lenses require an additional compensation for lens distortion [35], which is usually divided into a radial component and a de-centering or tangential component, approximated by a polynomial. Let  $(x, y)$  be a 2D image point without distortion. With  $r = \sqrt{x^2 + y^2}$ , the distorted point  $(x_d, y_d)$  is then retrieved by:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} x + (x - c_x)[r_1 r^2 + r_2 r^4 + \dots] + x_{tan} \\ y + (y - c_y)[r_1 r^2 + r_2 r^4 + \dots] + y_{tan} \end{pmatrix} \quad (7)$$

$$x_{tan} = [t_1(r^2 + 2(x - c_x)^2) + 2t_2(x - c_x)(y - c_y)](1 + t_3 r^2 + \dots)$$

$$y_{tan} = [2t_1(x - c_x)(y - c_y) + t_2(r^2 + 2(y - c_y)^2)](1 + t_3 r^2 + \dots)$$

where  $r_1, r_2, \dots$  and  $t_1, t_2, \dots$  are the radial and tangential distortion coefficients respectively. In the literature, there is no consensus about the number of coefficients that are necessary for perspective cameras with distortion. For example [21] uses two parameters each, while [61] uses only one coefficient for radial distortion and none for tangential distortion. Two coefficients for radial distortion



**Fig. 3.** Approximation of the underwater camera by the perspective model. A virtual image plane is used in combination with larger radial distortion to image the point onto the same radial coordinate. Even though, the two 3D points lying on the same ray in water are projected to the same pixel using the underwater model, but onto different pixels using the perspective model.

and none for tangential distortion are used by Zhang in [66]. A description of a widely used toolbox for perspective camera calibration is found in [2]. For our own experiments we use [52] with two coefficients for both components.

When using the perspective model on underwater images captured through a glass port, a calibration based on above-water images is invalid underwater. Furthermore, the perspective model itself is invalid for underwater images due to the non-single view point. Despite that, focal length and distortion coefficients can be used to approximate the difference introduced by not modeling refraction explicitly. Figure 3 depicts this approximation using cylinder coordinates:  $(R_w^1, Z_w^1)$  is a 3D point in water, which would be imaged to  $r_{uncompensated}^1$  without any compensation causing a large error compared to the true image  $r$ . By using a virtual image plane, which is moved further away from the center of projection, a part of this error can be compensated ( $r_{persp}^1$ ). Stronger radial distortion  $r_{comp}^1$  can be used to eliminate the error ( $r = r_{persp}^1 + r_{comp}^1$ ). The second 3D point  $(R_w^2, Z_w^2)$  is imaged with a greater point-camera distance on the same ray in water and it immediately becomes obvious, that the required compensation by radial distortion  $r_{comp}^2$  differs from  $r_{comp}^1$  and is therefore depending on the camera-point distance, a feature not supported by the common pinhole camera model. Hence, the approximation can only be satisfying for the calibration distance.

In spite of these problems, such an approximation offers the possibility of calibrating a camera above water and compute its approximate calibration for the underwater scenario, which is examined in the following two presented methods. Freyer et al. [12] use the pinhole camera model (with 3 parameters for radial distortion and 2 for tangential distortion) and compensate for refraction by multiplying the focal length in water by 1.34. More important in their opinion, is the change in the distortion parameters. When submerging a camera in water, the

change in radial distortion is specified to be  $\delta r = \left( \frac{\cos \theta_w}{\cos \theta_a} - 1 \right) r$ , with  $r$  being the radial distortion in air,  $\theta_w$  being the angle between optical axis and water ray, and  $\theta_a$  being the angle between optical axis and air ray. In common applications for perspective cameras, those angles are usually unknown.

Lavest et al. published a similar work in [30]. The paper explicitly models a thick lens, directly emerged in a medium other than air, which is then transferred into the pinhole model with distortion. Concerning the focal lengths in air and water, the major result matches the one introduced above:

$$1.333 f_{water} = f_{air}. \quad (8)$$

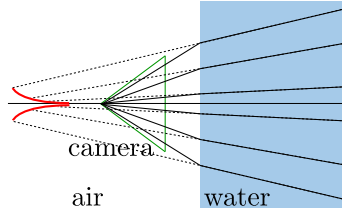
The computation of underwater distortion differs to the one in [12]: if  $r_{d_{air}}$  and  $r_{d_{water}}$  are the distorted coordinates in air and water respectively and  $r_{rad_{air}}$  and  $r_{rad_{water}}$  the corresponding radial distortion corrections, then

$$1.333(r_{d_{air}} - r_{rad_{water}}) = r_{d_{water}} - r_{rad_{water}}. \quad (9)$$

The authors experimented with two different cameras and their calibrations in air and water and found the theoretical equations (8) and (9) to be a good approximation. When considering the above discussion of figure 3, it becomes clear that unless  $r_{rad_{water}}$  is depending on the imaging distance, the model error is still not eliminated completely.

In case of using a dome port with a perfect fit, meaning the sphere's center coincides with the camera's center of projection, a calibration done in air is valid below water. According to the entry pupil model used for lens systems [1], the locus of the center of projection is determined by the lens system of the camera and can even lie in front of the physical camera and its lens. Consequently, it is a difficult task to perfectly align the camera center and the dome port's center. Alignment errors lead to even more complicated aberrations from the pinhole model than in the flat port case.

Despite of this usually inevitable geometric error, the literature contains a lot of methods (refer to tables 5,7), where the perspective model is used in underwater scenarios. Examples for calibrating a camera underwater are found in [4] or by Pessel et al. in [45,43,44]. Application areas utilize the implicitly contained geometric properties of the images to measure distances in stereo images [19,8], to compute dense stereo [51,39], to aid navigation by computing mosaics [15,16,13,5,63,9,60,42], or to reconstruct 3D structure (called Structure from Motion or SfM) [23,22,24,3,53,46,47,25,40,41]. The nature of these applications requires accurate geometric estimation. Especially the SfM approaches utilize navigation data, often available on a ROV (Remotely Operated Vehicle for underwater operations) in order to gain more accurately estimated camera poses and/or rely on extensive global optimization (bundle adjustment, refer to [59]). Otherwise, drift, i.e. an accumulating error in the recovery of the camera path, is a major problem sometimes causing the results to be useless. Some of the authors even mention the erroneous camera model as one of the error sources, but even though, up until now, most Structure from Motion approaches neglect the



**Fig. 4.** Radial image of a caustic (marked in red) caused by refraction at a water-air interface. When tracing the rays in water (dashed lines), they are tangents to the caustic.

error caused by refraction. Only recently, research has begun to explicitly incorporate refraction in a specialized SfM-system [6]. Section 5.3 will try to answer the question as to how severe the introduced error is for the applications.

### 3 Ray-Based Generic Camera Models

A possibility to account for refraction in underwater imaging more explicitly is to use a more generic camera model in underwater scenarios. Such ray-based cameras do not need to have a single viewpoint and are capable of dealing with dome ports and flat ports alike.

Grossberg et al. [17] introduce a generic camera model, where incoming rays are 'somehow' captured by corresponding pixels on the sensor. It is assumed that each pixel records exactly one main ray, no matter where on the ray the sensor array is. Therefore, the central definition of the paper, the raxel, describes one ray per pixel. When parameterizing all rays of an imaging system, there is usually a singularity in the bundle of rays (not true for e.g. orthographic cameras). The locus of this singularity is the caustic (fig. 4), uniquely describing the imaging system. In case of a single view point system, the caustic encompasses only a single point - the center of projection.

In order to compute the caustic, the mapping from image coordinates to rays is differentiated.

$$\mathbf{X}(x, y, \alpha) = \begin{pmatrix} X(x, y, \alpha) \\ Y(x, y, \alpha) \\ Z(x, y, \alpha) \end{pmatrix} = \mathbf{X}_s(x, y) + \alpha \tilde{\mathbf{X}}(x, y), \quad (10)$$

with  $\mathbf{X}_s$  being the starting point and  $\tilde{\mathbf{X}}$  being the direction of the ray starting at image point  $(x, y)$  and  $\alpha \in \mathbb{R}$  describing the position on the ray. The determinant of the Jacobi matrix of this parametrization is set to zero and solved for the parameter  $\alpha$ .

$$\det(J(\mathbf{X}(x, y, \alpha))) = 0 \quad (11)$$

Using  $\alpha$  in (10) allows to compute the corresponding point on the caustic for each pixel position  $(x, y)$ . Grossberg et al. develop a method to compute

caustics for arbitrary cameras numerically by projecting differing calibration patterns using an active display. Unfortunately such active displays are not feasible in underwater environments, but as was noted by [58] (see also below), caustics provide a natural connection between the underwater non-single-view-point camera, generic ray-based cameras, and the common pinhole model.

A different work by Narasimhan et al. [38] researches light sheet reconstruction as an application of the described raxel model for small scale underwater images in laboratory settings. A camera is put in front of a water tank, and calibrated by placing two planes into the tank vertically with respect to the optical axis and therefore gaining two points in space for each ray.

In addition to the raxel model, Sturm et al. [55,54] work with another ray-based model, where each pixel is simply represented by a ray defined by its starting point and its direction of travel. By only assuming that neighboring rays are close to each other, this model is independent of the physical location of the sensor array and does not require an existing caustic, thus making the camera model even more generic than the raxel model. A camera is calibrated by taking several images of a calibration plane, however, the authors mention problems with the calibration robustness. In [54], algorithms for pose estimation, triangulation, multi-view geometry, in short for SfM, are derived and the theory is applicable to the underwater case. [7] concentrates on the case of a refractive plane in an underwater scenario. The derivation only works for one refractive interface (thin glass) and has not yet been implemented.

Another possibility to deal with refraction by approximating ray-based cameras is described in [62]. Here, the camera is viewed as a non-SVP camera having a caustic instead of the single view point. Instead of modeling the refraction effect physically or using a generic ray-based camera, the camera is approximated by several perspective cameras for the different areas of the image. The number of virtual perspective cameras determines the accuracy of this system.

In summary, it can be said, that using a more generic camera model than the pinhole model with distortion allows to deal with refractive effects. However, using independent 3D origins and directions for each ray leads to a high degree of freedom, making the robust calibration of generic camera models difficult, especially in open water. The following section shows that far less parameters need to be determined if refraction is modeled explicitly.

## 4 Physical Models for Refraction

The third possibility to deal with refraction is to use a physical model that explicitly computes the refraction of rays at the underwater housing. Several methods for achieving this will be compared in this section. They differ in the assumptions made about the glass thickness, normal between interface and image sensor plane, or indexes of refraction and in their derivation.

The two papers presented next describe the theory and calibration method for calibrating underwater cameras with the assumption of a thin piece of flat glass as an interface of the underwater housing.

In [58] by Treibitz et al., the derivation of a refractive model and its calibration for a perspective camera behind a flat port are presented. The authors' underwater housing has a glass thickness of about  $5\text{mm}$ . The ray's shift due to traveling through the glass interface is approximated to be about  $0.28\text{mm}$  and therefore neglected. In addition, it is assumed that the image sensor and the interface are parallel. This allows examining the projection through a refractive interface by using radial image coordinates, thus making it possible to derive all required equations analytically.

The derivation is based on Fermat's principle (see [1.1]):

$$\frac{dt}{dR_i} = n_w \frac{-(R_w - R_i)}{\sqrt{(Z_w - d)^2 + (R_w - R_i)^2}} + n_a \frac{R_i}{\sqrt{d^2 + R_i^2}} = 0, \quad (12)$$

where,  $(R_i, d)$  is the radial coordinate on the interface and  $(R_w, Z_w)$  is the radial coordinate of the 3D point in the water. For common perspective systems with only small amounts of radial distortion, the following equation holds for all radial coordinates:

$$f \approx \frac{Z_w r}{R_w}, \quad (13)$$

with  $f$  being the focal length. This equation can be used to project (radial) coordinates on the glass interface  $(R_i, d)$  into the perspective camera:

$$R_i = rd/f \quad (14)$$

Using this in equation (12) yields:

$$\left(R_w - \frac{d}{f}r\right)^2 \left[\left(\frac{fn_w}{r}\right)^2 + (n_w^2 - 1)\right] = Z_w^2 \quad (15)$$

relating  $r$  and  $(R_w, Z_w)$  in the underwater case. In order to calibrate the camera model, the common parameters for perspective cameras ( $f$ ,  $(c_x, c_y)$ ,  $r_1$ ,  $r_2$ ) as well as the interface distance  $d$  are calibrated. Based on (15), the following equation needs to be satisfied:

$$R_w = \frac{Z_w}{\sqrt{\left(\frac{fn_w}{r}\right)^2 + (n_w^2 - 1)}} + \frac{d}{f}r_i \quad (16)$$

which is extended to account for lens distortion. When using two points  $\mathbf{X}_{w_1}$  and  $\mathbf{X}_{w_2}$ , they are parametrized by  $(R_{w_i}, \varphi_{w_i}, Z_{w_i})$  and their distance in space is estimated using the law of cosines:

$$\hat{s} = \sqrt{\hat{R}_{w_1}^2 + \hat{R}_{w_2}^2 - 2\hat{R}_{w_1}\hat{R}_{w_2}\cos|\varphi_{w_1} - \varphi_{w_2}|}. \quad (17)$$

Using the true distances  $s$ , a non-linear optimization is used to solve for the camera parameters. Usually, the intrinsic parameters apart from focal length are

estimated beforehand, leaving only the interface distance and the focal length to be calibrated.

As introduced by [17], caustics can be used as a measure of the deviation from the single view point model. [58] derives the caustic analytically using equation (16) (see section 3):

$$R_{caustic} = \left(1 - \frac{1}{n_w^2}\right) \left(\frac{r}{f}\right)^3 d \quad (18)$$

$$Z_{caustic} = -n \left[1 + \left(1 - \frac{1}{n_w^2}\right) \left(\frac{r}{f}\right)^2\right]^{1.5} d \quad (19)$$

Obviously, the caustic's extent is directly depending on the interface distance  $d$ , therefore, the extent of the caustic can be diminished by moving the entry pupil as close to the glass interface as possible.

Telem et al. describe in [56,57] a different approach to model refraction. As in the approach described above, the authors use a model with thin glass and parallelism between image sensor and interface in the first paper, but in their photogrammetric model, the authors relate the measured 2D image coordinates to image coordinates eligible for perspective projection. Note that the camera center is not valid for these points, so the intersection with the ray coming from the water and the optical axis is computed for each set of image coordinates as well. The non-refracted rays do not meet in one common center of projection. For each point (in radial coordinates) a value

$$\Delta d = d \left( \frac{n_w}{n_a f} \sqrt{f^2 + r^2 \left(1 - \frac{n_a^2}{n_w^2}\right)} - 1 \right) \quad (20)$$

specifies the distance between the center of projection and the actual crossing of the non-refracted ray with the optical axis. The measured image points  $(x, y, f)$  are modified in the underwater case to fit the perspective projection depending on  $\Delta d$ , the water ray's crossing with the optical axis:

$$\begin{pmatrix} x' \\ y' \\ f' \end{pmatrix} = \begin{pmatrix} x \frac{d}{f} \\ y \frac{d}{f} \\ d + \Delta d \end{pmatrix}. \quad (21)$$

This allows to write the back projection to an underwater ray as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = (\mathbf{C} + \Delta \mathbf{C}) + \lambda \mathbf{R}^T \begin{pmatrix} x' \\ y' \\ f' \end{pmatrix} \quad (22)$$

with  $\Delta \mathbf{C} = -\Delta d \mathbf{r}_3$  being the deviation from the principle point and  $\mathbf{r}_3$  being the third row of the rotation matrix  $\mathbf{R}$ . In a second paper [57], the authors extend their approach to incorporate glass interfaces that are not parallel to the image sensor, causing  $\Delta d$  to become more complicated. However, in our tests, we found



that the ray coming from the water not necessarily intersects the optical axis if the interface is not parallel to the image sensor. Errors introduced by a non-zero glass thickness are absorbed by the interface distance. In the calibration tests, the intrinsics are estimated first, then four additional parameters for the underwater case are calibrated:  $d$ ,  $n = n_w/n_a$ , and two parameters for the interface rotation. The results show that it is possible to estimate the required underwater parameters and the missing camera poses without getting large correlations between the parameters.

An often cited method [28,29] establishes a way to combine refraction with the pose estimation using the Direct Linear Transform (DLT [18]). However, parallelism between interface and image sensor was achieved by manually rotating the hardware, and the distance between interface and camera center is measured. The authors have so far not included an estimation of those parameters into their algorithm.

Up until now, all described methods considering a physical refraction model assumed thin glass and, except for one, parallelism between interface and image sensor. Li et al. [32,31] (see also [35]) describe an approach based on back projecting image points, with a stereo rig where the complete physical model is calibrated: the double refraction of rays at the air-glass and the glass-water interfaces is modeled explicitly. Here, the light is assumed to travel through  $p + 1$  different refractive media and thus is refracted  $p$  times. This is derived using Snell’s law instead of Fermat’s principle: the points  $(X_i, Y_i, Z_i)$  and  $(X_{i-1}, Y_{i-1}, Z_{i-1})$  denote the points on the  $i$ -th and  $i-1$ -th interface surfaces. The path length of the ray between those interfaces is:

$$\rho = \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2 + (Z_i - Z_{i-1})^2}. \tag{23}$$

In addition, it is assumed that the start and end points  $(X_0, Y_0, Z_0)$  and  $(X_{p+1}, Y_{p+1}, Z_{p+1})$  are known as well as the functions of the refractive surfaces  $F_i(X_i, Y_i, Z_i) = 0$  with their existing derivation:  $\left[ \frac{\partial F_i}{\partial X_i}, \frac{\partial F_i}{\partial Y_i}, \frac{\partial F_i}{\partial Z_i} \right]^T$ .

Using those notations, Snell’s law is applied at each refractive point:  $n_i \sin \theta_i = n_{i+1} \sin \theta'_i$  and the ray between the interfaces is determined by:

$$\tilde{\mathbf{X}}_i = \frac{1}{\rho_i} \begin{pmatrix} X_i - X_{i-1} \\ Y_i - Y_{i-1} \\ Z_i - Z_{i-1} \end{pmatrix}. \tag{24}$$

With  $\mathbf{n}_i$  being the normal at the interface point (computed using the derivatives of function  $F_i$ ),  $\theta_i$  and  $\theta'_i$  are computed using the scalar product:

$$\cos \theta_i = \mathbf{n}^T \tilde{\mathbf{X}}_i \qquad \cos \theta'_i = \mathbf{n}^T \tilde{\mathbf{X}}_{i+1}, \tag{25}$$

allowing the computation of the following function using Snell’s law and the fact that both rays and the normal form the same plane:

$$\tilde{\mathbf{X}}_{i+1} = \frac{n_i}{n_{i+1}} \tilde{\mathbf{X}}_i - \left( \frac{n_i}{n_{i+1}} \cos \theta_i - \cos \theta'_i \right) \mathbf{n}_i. \tag{26}$$

Using (26), inner interface points are back projected, then refracted twice resulting in outer interface points and rays in water eligible for triangulation using the stereo rig. The calibration routine assumes known indexes of refraction and estimates the intrinsics and rig extrinsics from images taken in air. Then the underwater parameters are calibrated by taking images of a three-dimensional calibration object underwater using linearized versions of the equations derived above to find an initial solution. The accuracy evaluation in [32] showed that the errors of reconstructed 3D points are between  $6mm$  and  $6cm$  for the optical axis and  $6mm$  and  $1cm$  for the x- and y-axes. In [31], an additional reduced central projection allows to project points from 3D through a refractive interface onto the image plane with an iterative method that solves for the required unknowns on the interfaces.

In [27], the usage of a perspective camera in an underwater scenario is examined as well as a flat port and dome port model. The back projection is derived by computing rays in air, glass, and water using Snell's law and quaternion rotations (refer to section 5.1). Projections are computed numerically. In addition, a calibration routine is proposed assuming intrinsics, indexes of refraction, and glass thickness to be known. Nested loops of a Levenberg Marquardt routine [48] are used to solve for the remaining interface parameters and the camera's poses with respect to a calibration pattern. Unfortunately, the authors did not implement and examine their calibration routine, but conclude that consideration of refraction is necessary when exploring the implicitly contained geometric information from images due to the model error (see section 5). Chang and Chen [6] made a promising start in developing an actual 3D-reconstruction algorithm with explicit consideration of refraction. The cameras are assumed to view the object of interest through the planar water surface. The vertical direction of the camera is assumed to be known, so only the heading of the camera needs to be computed.

Another approach to using physical models is found in the works of Maas, [33,34] and a follow-up work by Putze [50,49]. The goal of both methods is optical fluid flow analysis in fairly small laboratory tanks, where the fluid has been marked with a set of particles. In the model, the actual 3D points in space are substituted by their corresponding virtual 3D points, fitting the perspective back projection. The computation of these points is based on an iteration with known interface parameters and indexes of refraction. In order to calibrate the system, a calibration pattern below water at known distances is used and optimized by a bundle adjustment routine. The method has been found to perform well if the indexes of refraction, especially for the glass are known. A correlation analysis shows high correlation between focal length and distance between camera center and glass interface for all three calibrated cameras. The works of Maas also contain an introduction to epipolar geometry [18] in case of refractive imaging, where the epipolar lines are bent into curves. If the ray in water from one camera is known, several points on this ray are projected into the second image defining a linear approximation of the epipolar curve. This is for example used in [11] examining surface reconstruction.

In addition, there exist some more exotic applications also considering refraction explicitly. In contrast to the approaches described above, where the indexes of refraction are assumed to be known, here, they can be calibrated only in very confined laboratory scenarios. See [37,64,65,26,10] for more detailed information.

The methods for using a physical model of refraction on underwater images show that calibrating such systems is possible only if extra assumptions about interface-sensor parallelism, indexes of refraction, or glass thickness are made or a stereo rig is used. Until now, methods utilizing geometric information contained in images usually rely on the perspective camera model, but [27] already showed that a considerable error is caused by using the wrong camera model, however, we found that an inclination angle between housing interface and image sensor is even worse than different interface - camera distances. Therefore, the analysis in [27] will be extended in the following section.

## 5 Accuracy Analysis of the Perspective Model

In this section, the exact derivation of the physical underwater ray cast will be explained. This ray cast is then used to compute synthetic data compliant with the underwater model allowing to compute for example the caustic for the case of non-parallel interface and sensor plane. As shown in section 2, most authors still work using a perspective camera on underwater images and this section aims at examining the resulting error and its compensation in detail by using the synthetic data computed by physically modeling refraction.

### 5.1 Physical Underwater Projection

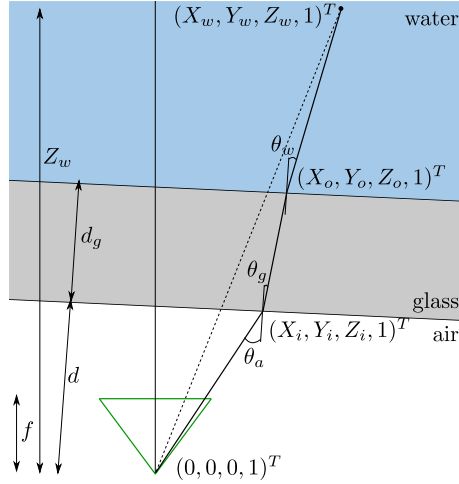
The derivation of the ray cast in the physical underwater model presented here follows [27], but is more detailed and considers projection routines. Note that other papers using Snell's law for the derivation come to similar conclusions.

**Flat Port Back Projection** in case of a flat port in front of an underwater housing, the distance to the port, the glass thickness, and the normal of the glass within the camera coordinate system are important parameters. Here, the inner interface plane is parametrized by  $\Pi_i = (\underbrace{n_1, n_2, n_3}_{\mathbf{n}_\Pi}, -d)$  containing

the normal and the port's distance to the camera origin. In addition, the outer interface plane is parametrized by the same normal and the glass thickness  $d_g$ :  $\Pi_o = (n_1, n_2, n_3, -(d+d_g))$  (fig. 5). When back projecting an image point in the underwater case, the goal is the computation of the point on the outer interface plane and the direction of the ray within the water. First, the image point  $\mathbf{x}$  needs to be turned into a ray within the camera's underwater housing:

$$\tilde{\mathbf{X}}_a = \mathbf{K}^{-1}\mathbf{x}, \quad (27)$$

with the subscript  $a$  denoting the coordinates within the underwater housing, in air, and  $\mathbf{K}$  being the camera matrix containing the intrinsic parameters. The ray is in the camera coordinate system, i.e. the center of projection is in the origin.



**Fig. 5.** When back projecting a point (solid line), the ray travels from the camera through air until it intersects the inner interface plane  $(X_i, Y_i, Z_i, 1)^T$ . After being refracted, the ray travels through glass until intersecting the outer interface plane  $(X_o, Y_o, Z_o, 1)^T$ , is then refracted, and finally travels through water reaching the 3D point in water  $(X_w, Y_w, Z_w, 1)^T$ . Projecting  $(X_w, Y_w, Z_w, 1)^T$  without refraction (dashed line) yields a different pixel in the image.

In order to find the intersection  $\mathbf{X}_i$  between ray and interface the following equation is used:

$$\Pi_i^T \begin{pmatrix} \lambda_g \tilde{X}_a \\ \lambda_g \tilde{Y}_a \\ \lambda_g \tilde{Z}_a \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \lambda_g = \frac{d}{\mathbf{n}_{II}^T \tilde{\mathbf{X}}_a} \quad \Rightarrow \mathbf{X}_i = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \lambda_g \tilde{\mathbf{X}}_a. \quad (28)$$

The intersection of the port's inner plane and the ray, parametrized by  $\lambda_g$ , is used to determine the point on the inner plane of the interface  $\mathbf{X}_i$ . After that, the ray's direction within the glass is computed: the scalar product between the plane's normal  $\mathbf{n}_{II}$  and the ray in air yields the angle between normal and incident ray before refraction, and is then refracted:

$$\theta_a = \arccos \left( \frac{\mathbf{n}_{II}^T \tilde{\mathbf{X}}_a}{\|\mathbf{n}_{II}\| \|\tilde{\mathbf{X}}_a\|} \right) \quad \theta_g = \arcsin \left( \sin \theta_a \frac{n_a}{n_g} \right). \quad (29)$$

The ray being incident upon the inner interface plane needs to be rotated/refracted. This is described by a rotation around the normal resulting from the cross product of the plane normal and the incoming ray:

$$\mathbf{n}_{rot} = \frac{\mathbf{n}_{II} \times \tilde{\mathbf{X}}_a}{\|\mathbf{n}_{II}\| \|\tilde{\mathbf{X}}_a\| \sin \theta_a}. \quad (30)$$

The rotation angle is  $\theta_{rot} = \theta_g - \theta_a$  and the rotation around the axis  $\mathbf{n}_{rot}$  is best described by a unit quaternion:

$$\mathbf{q} = \begin{pmatrix} \sin\left(\frac{\theta_{rot}}{2}\right) \\ \frac{\sin\left(\frac{\theta_{rot}}{2}\right)}{\|\mathbf{n}_{rot}\|} \mathbf{n}_{rot} \\ \cos\left(\frac{\theta_{rot}}{2}\right) \end{pmatrix}. \quad (31)$$

This quaternion is applied to the ray  $\tilde{\mathbf{X}}_a$ , yielding the refracted ray  $\tilde{\mathbf{X}}_g$ , which describes the light's traveling direction within the glass. Now, the point on the outer interface needs to be computed:

$$\lambda_w = \frac{(d_g + d - \mathbf{n}_{II}^T \mathbf{X}_i)}{\mathbf{n}_{II}^T \tilde{\mathbf{X}}_g} \quad \Rightarrow \quad \mathbf{X}_o = \mathbf{X}_i + \lambda_w \tilde{\mathbf{X}}_g. \quad (32)$$

The ray within the glass is refracted again, using the indexes of refraction for glass and water, the cross product, and the unit quaternion rotation. The result is the ray in water  $\tilde{\mathbf{X}}_w$ . The 3D point can be computed, if the distance *dist* between the camera center and the 3D point is known:

$$\|\mathbf{X}_o + \alpha_w \tilde{\mathbf{X}}_w\| = dist \quad (33)$$

This equation can be solved for  $\alpha_w$  yielding the distance the ray needs to travel from the interface point:

$$\mathbf{X}_w = \mathbf{X}_o + \alpha_w \tilde{\mathbf{X}}_w. \quad (34)$$

$\mathbf{X}_w$  is still in the camera coordinate system, but using the transform of the camera pose, the point can easily be transformed into the world coordinate system.

**Dome Port Back Projection** the dome is parametrized by its center with respect to the camera's center of projection and its inner and outer radius. In case of perfect alignment of the dome center and the camera's center of projection, the project and back project functions are equal to the common pinhole camera model. Otherwise, the refraction at the dome needs to be modeled explicitly, but the only difference to the method described above is found in the intersection of the rays in air or glass and the inner and outer interface respectively. To compute the intersection point, the inner and outer dome spheres are parametrized by using the quadric:

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (35)$$

A transformation containing the sphere's inner  $r_i$  or outer  $r_o$  radius and the translation of the dome's center  $(X_d, Y_d, Z_d)^T$  are applied to the quadric to get the matrix describing the dome:

$$\mathbf{H}_i = \begin{pmatrix} r_i & 0 & 0 & X_d \\ 0 & r_i & 0 & Y_d \\ 0 & 0 & r_i & Z_d \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{36}$$

$$\mathbf{D}_i = (\mathbf{H}^{-1})^T \mathbf{Q} \mathbf{H}^{-1}.$$

A homogeneous point  $\mathbf{X}$  lies on the quadric  $\mathbf{D}$  if  $\mathbf{X}^T \mathbf{D} \mathbf{X} = 0$ . Using the parametrization for the ray in air or in glass, the intersections of the rays with the inner or outer dome surface can be determined. The normals at those intersection points can be found by using the line from the center of the dome to the intersection points. Once the normals, the intersection points, and the ray directions in air and glass are known, the remaining derivation of the refraction is exactly the same as in the flat port case.

**Projection** in contrast to [27], we analyze the projection of 3D points into the camera in more detail, using an approach building upon [58]. The problem in this case is caused by the unknown points on the inner and outer interface. In order to derive a formula for the projection, Fermat’s principle is applied. The total traveling time of the ray is the sum of three components: the time spent in the underwater housing (in air), the time spent in the glass of the interface, and the time spent in the water. The derived equation contains four unknowns, the x- and y-coordinates on the inner and outer interface planes ( $X_i$  and  $Y_i$  and  $X_o$  and  $Y_o$  respectively):

$$t(X_i, Y_i, X_o, Y_o) = \tag{37}$$

$$n_{air} \sqrt{X_i^2 + Y_i^2 + Z_i^2} +$$

$$n_{glass} \sqrt{(X_o - X_i)^2 + (Y_o - Y_i)^2 + (Z_o - Z_i)^2} +$$

$$n_{water} \sqrt{(X_w - X_o)^2 + (Y_w - Y_o)^2 + (Z_w - Z_o)^2}.$$

Since the light always travels the distance in the least time, this equation’s partial derivatives are used to minimize the traveling time with respect to the unknowns:

$$\frac{\partial t}{\partial X_i} = 0 \qquad \frac{\partial t}{\partial Y_i} = 0 \qquad \frac{\partial t}{\partial X_o} = 0 \qquad \frac{\partial t}{\partial Y_o} = 0. \tag{38}$$

The plane equations are utilized to eliminate the  $Z$ -components:

$$Z_i = \frac{d - n_1 X_i - n_2 Y_i}{n_3} \tag{39}$$

$$Z_o = \frac{d + d_g - n_1 X_o - n_2 Y_o}{n_3}.$$

The resulting system of equations with four equations and four unknowns is solved numerically using e.g. Powell’s hybrid method [48]. After that, the

<sup>1</sup> e.g. in GSL library from [www.gnu.org/software/gsl/](http://www.gnu.org/software/gsl/)

**Table 3.** Parameters used for caustic computation

focal length	1100 px
image size	1001 × 801 px
principal point	middle of image
distortion	$r_1 = 0, r_2 = 0, t_1 = 0, \text{ and } t_2 = 0$
aspect ratio	1
skew	0
index of refraction water	1.333
index of refraction glass	1.5
index of refraction air	1
interface distance	20mm
glass thickness	30mm
interface tilt	1.5°

points on the inner and outer interface planes are determined, however, only the point on the inner interface plane is relevant for projecting it onto the image plane with the usual perspective projection.

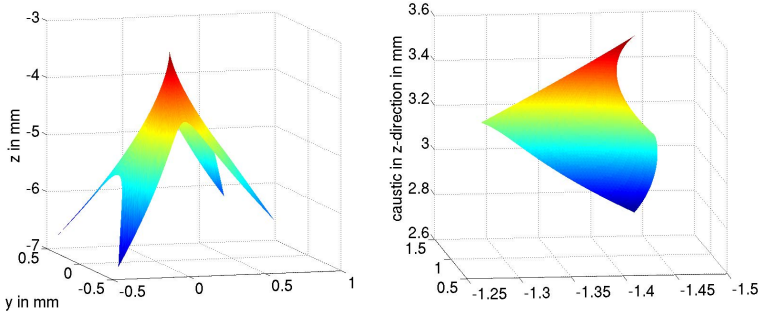
In our tests, we found that it is difficult to find the correct solution using this method, especially in case of a negative camera-interface distance. This occasionally happens, when the entry-pupil of the camera lies in front of the physical camera housing (refer to [1,58]). In case of thin or no glass, parallelism between interface and image sensor and positive interface distance  $d$ , (38) is only depending on the radial coordinate on the refractive plane. The derivative in this direction becomes a polynomial of fourth degree [14,58]. For this special case, [14] proved that the correct/practical root is found in the interval  $[0, R_w]$ . In experiments in our more general case, with possibly negative  $d$  and non-parallel interface, this is no longer true. In order to deal with all possible cases, the projection can also be solved numerically (as in [27]). This is accomplished by an optimization, which is initialized using the common perspective projection. After that, the Nelder-Mead-Simplex routine<sup>2</sup> [48] is used to compute the correct 2D point.

## 5.2 Caustics as a Measure of Deviation from the SVP

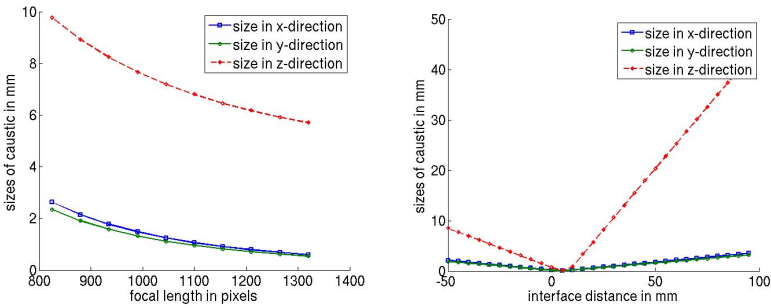
Caustics present the bridge between physically modeled underwater cameras and more generic camera models. The extent of a caustic is also a measure of the deviation from the perspective single view point camera. [17,62,58] describe methods for deriving caustics analytically. In more generic models with thick glass and no parallelism between the sensor and the interface, the analytic derivation becomes infeasible.

Alternatively, the outer interface points and directions of the ray in water are computed using the back project function described above. The derivatives for the Jacobi matrix are computed numerically.  $\alpha$  (parameterizing the point

<sup>2</sup> NLOPT toolbox from [ab-initio.mit.edu/nlopt/](http://ab-initio.mit.edu/nlopt/)



**Fig. 6.** Left: the caustic for a flat port camera housing with imperfect sensor-interface alignment. Right: caustic in the dome port case with imperfect alignment.



**Fig. 7.** Left: caustic size depending on focal length for  $30\text{mm}$  glass thickness and interface tilt =  $1.5^\circ$ . Right: caustic size depending on interface distance for  $30\text{mm}$  glass thickness and interface tilt =  $1.5^\circ$ .

on each ray, which lies on the caustic) is expressed in terms of the entries of the Jacobian. Once  $\alpha$  is known, the ray parametrization is used to compute the caustic point for each image point  $(x, y)$ . Figure 6 shows an exemplary caustic for the parameters in table 3, and figure 7 is an example for the extent in x- y- and z-direction, which changes with focal length and distance between camera center and interface, and can be in the order of centimeters.

### 5.3 Accuracy of the Perspective Model in Calibration, Triangulation, and SfM

In this section, results of the accuracy analysis of using the perspective model on underwater images from [58, 27] are extended, especially considering slight rotations of the interface plane.

**Error Compensation in Perspective Calibrations.** Using an implementation of the model described above, a thorough examination based on synthetic



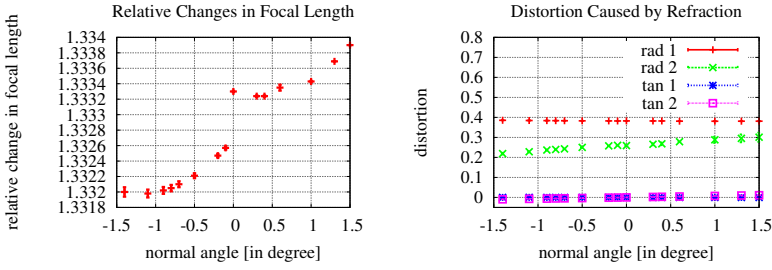
**Table 4.** Parameters used for synthetic tests

focal length	800,1000 px
image size	600 × 800 px
principal point	middle of image
distortion	no distortion
aspect ratio	1
skew	0
index of refraction water	1.333
index of refraction glass	1.45
index of refraction air	1
interface distance	10 - 80 mm
glass thickness	5, 30, 60 mm
interface rotation	0° or 2 - 3°
rig baseline (no rotation)	200 mm
distance range checkerboard images	1000-10000 mm

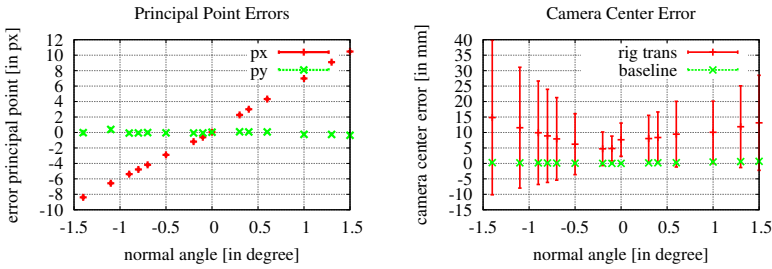
data is possible. The synthetic images were rendered according to the underwater projection model for a stereo camera rig. In order to examine the influence of different underwater housings, different sets of calibration images (50 for each set) showing a checkerboard pattern were rendered with different parameters. Using the exact checkerboard corners to eliminate effects from erroneous corner detection, the camera rigs were calibrated perspectively using [52]. Table 4 summarizes the parameters for different test cases.

When using the error-free 2D3D correspondences from perspective projections for calibration in [52], the final re-projection error is in the order of  $10^{-8}$  (model and data fit perfectly). When using 2D3D correspondences compliant with the underwater model, the final re-projection error is in the order of ( $\varnothing < 0.05$  pixel), which still suggests a good fit to the perspective model. As stated by [12,30], the focal length changes according to the refractive index of water when calibrating perspectively, see figure 8 on the left. The underwater images were rendered without any distortion, so the four resulting parameters (fig. 8, right) give an idea about how much the images are altered by refraction. Obviously, tangential distortion does not compensate the error induced by tilting the interface. Figure 9 on the left shows the resulting errors in principal points of the calibration. In the case of a slightly rotated interface plane, part of the error is absorbed by moving the principal point. Furthermore, the computed camera centers have an increasing error and increasing covariances (see figure 9 on the right), not only suggesting problems with robustness, but an error in the extrinsic parameters during the calibration causes errors during applications later on.

In case of dome ports, [27] came to the conclusion that perspective models are accurate enough if the camera center does not move more than 1cm from the dome center.

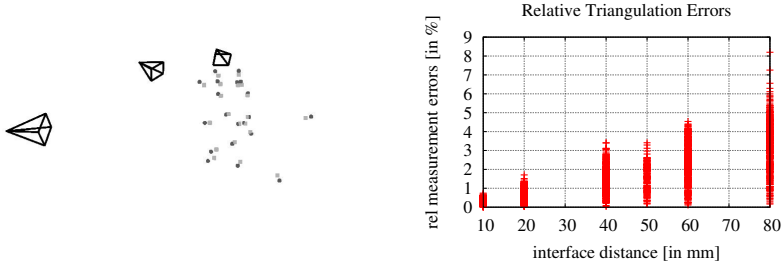


**Fig. 8.** Left: relative change in focal length when calibrating perspectively:  $\frac{f_{l_{persp}}}{f_{l_{water}}}$ . The true index of refraction for water was 1.333, thus, refraction is partly compensated for by using a virtual sensor plane. Right: distortion introduced by using the perspective calibration (camera within the underwater housing had zero distortion). Plotted are the four used coefficients for distortion: two each for radial and tangential distortion. Radial distortion mainly compensates for refraction in general, and tangential distortion does not compensate a rotation of the interface plane.

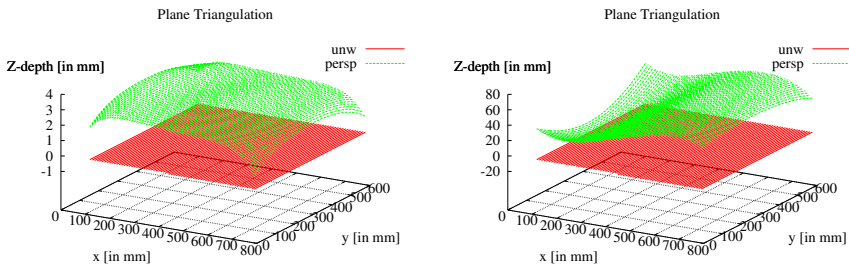


**Fig. 9.** Left: errors of estimated principal points in pixels depending on the interface angle. Right: errors of estimated camera centers in mm depending on the interface angle.

**Triangulation Errors.** When using a perspectively calibrated underwater camera for tasks such as measuring using stereo rigs or computing 3D reconstructions, accuracy and drift reduction play important roles. The error induced by using the perspective model for triangulating points is shown in figure 10. The left image shows a rendering of 3 cameras and 20 triangulated points. In dark gray are the true points triangulated using the underwater model, while the light gray points were triangulated using the perspective model. It can be seen clearly that the perspective calibration has an area where it fits well, allowing fairly accurate reconstruction, while in other areas of the 3D space high triangulation errors occur. The right figure 10 shows triangulation errors depending on the interface distance for the stereo rig calibrated in different parameter configurations. In addition to the

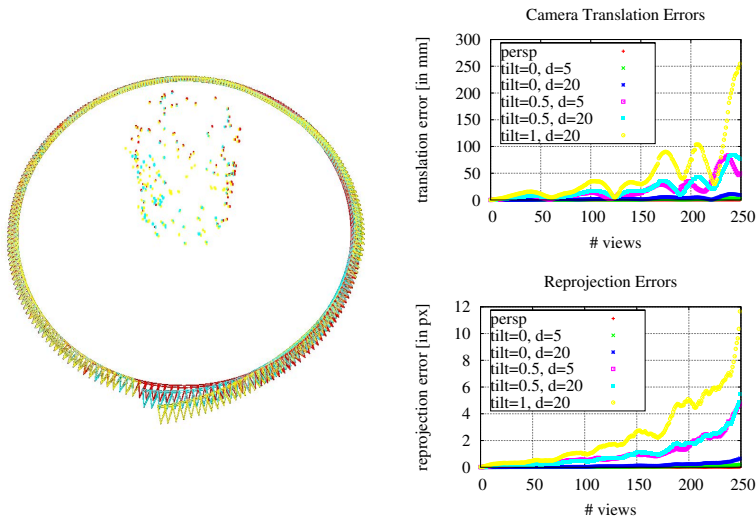


**Fig. 10.** Left: triangulation results after triangulating with three cameras. Dark gray: correct points triangulated with physical model for underwater camera, light gray: points triangulated with perspective calibration of the underwater images. Right: relative triangulation errors using the perspective calibrations depending on the interface distance (relative with respect to camera-point distance in %).



**Fig. 11.** Two of the perspective calibration scenarios, both with interface distance  $20\text{mm}$  were used to triangulate points on the  $xy$ -plane, with the camera being  $2\text{m}$  away, viewing the  $xy$ -plane at an  $45^\circ$  angle. The left scenario has a parallel interface with respect to the imaging sensor, while in the right scenario the interface was tilted by  $(-1^\circ, 1^\circ)$  with the resulting errors in the perspective calibration. 'persp' stands for perspective triangulation, while 'unw' stands for refractive triangulation. Note the different scales of the  $z$ -axis.

dependence on the interface distance, the error also depends on the distance of the points used for triangulation to the center of projection and the distance range of the calibration pattern with respect to the camera. Figure 11 extends the analysis of plane triangulation in [27] by comparing parallel and non-parallel interfaces: the black (red in color version) planes are triangulated using the underwater model (interface distance  $20\text{mm}$ , glass thickness  $30\text{mm}$ ), while the gray (green in color version) planes are triangulated using the perspective calibration. In case of parallelism between interface and image sensor, the error is radially symmetric (11, left), while in the right image, a slight rotation of the interface plane causes far higher errors.



**Fig. 12.** The reconstructed points in the middle all lie on a cylinder (scene size = 2500mm). In dark gray (red in the color version) are the cameras and points in the perspective scenario. In medium gray (cyan in the color version) ( $tilt = 0.5^\circ$ ,  $d = 20$ ) and light gray (yellow in the color version) ( $tilt = 1^\circ$ ,  $d = 20$ ) are the points and cameras, when the points originate from the underwater model, while a perspective calibrated camera is used to compute the reconstruction using a classical SfM approach. On the right, error curves for the different scenarios are shown.

**Errors in Pose Estimation.** Figure 12 shows the reconstruction of a cylinder captured from cameras moved on an orbit with slight interface rotation. The error induced by the wrong camera model clearly accumulates. Note that the correspondences used are synthetic and therefore not biased by feature detection and matching methods, so all of the drift in this scenario is caused by the model error alone, increasing especially in case of even slight rotations between interface and image sensor.

Aside from other sources of error not present in the synthetic data presented here (e.g. errors in checkerboard detection), the measurement errors induced by using an incorrect imaging model do not bode well for exact measurements of underwater structures. This matches the conclusions drawn in [27,6]: underwater SfM so far works even in case of several thousand images, however, navigational data or time consuming bundle adjustment methods are required to stabilize the motion computation and reduce drift.

Table 5. Paper overview

Authors	Application	Method
<b>Perspective Model</b>		
Freyer et al. [72]	calibration	calibrate in air and find perspective water calibration by adapting focal length ( $f_{water} \cdot 1.333 = f_{air}$ ) and distortion ( $\delta r = \left( \frac{\cos \theta_w}{\cos \theta_a} - 1 \right) r$ , with $r$ being the radial distortion in air, $\theta_w$ being the angle between optical axis and water ray, and $\theta_a$ being the angle between optical axis and air ray)
Lavest et al. [30]	calibration	calibrate in air and find perspective water calibration by adapting focal length ( $f_{water} \cdot 1.333 = f_{air}$ ) and distortion ( $1.333u_{air} + du_{air} = u_{water} + du_{water}$ , with $u_{air}$ and $u_{water}$ being the distorted coordinates and $du_{air}$ and $du_{water}$ being the distortion corrections)
Bryant et al. [4]	calibration	finding checkerboard corners robustly in turbid environments; calibrate based on underwater images; using one coefficient for radial lens distortion
Pessel et al. [45][43][44]	calibration	checkerboard-free self calibration approach using predefined trajectory; no distortion modeled because lens system eliminated distortion effects by 99%; calibrate on-site to adapt calibration to changing index of refraction of water
Harvey et al. [19]	stereo measurement	usage of stereo rig to measure underwater structures; examination of calibration robustness in different water bodies; 3D calibration frame
Costa et al. [8]	stereo measurement	automatically measure fish size using a stereo rig; automatic contour detection and interest point triangulation; initial calibration without distortion, removal of inconsistencies by training neural network; 5% measuring accuracy
Gracias et al. [15][16]	mosaicing	a mosaic computation; used for navigation after wards; self calibration with rotating camera on pan-tilt unit, sequential mosaic building followed by global optimization; geo-referenced
García, Carreras et al. [13][5]	mosaicing	a mosaic computation; used for navigation after wards; one parameter for radial distortion; second paper using robot in pool with coded pattern on the ground for estimating the accuracy of other on-board navigation devices
Xu, Negahdaripour et al. [42][63]	mosaicing	first: simultaneous mosaicing, navigation, and station keeping; second: statistical combination of image-based registration data and other navigation data sources applied to mosaic computation
Eustice et al. [9]	mosaicing	compares different methods for mosaicing under consideration of movement with growing complexity (from translation to full projective transformations) in underwater environments
Trucco et al. [60]	mosaicing	mosaicing approach via tracked features and homography estimation; registered images are warped into common image

Table 6. Paper overview

Authors	Application	Method
Hogue et al. [23,22]	3D reconstruction	a bumblebee stereo camera and IMU are combined in one underwater housing and used to reconstruct and register 3D structure; reconstruction shows a lot of drift if IMU is not used and authors presume erroneous camera model to cause part of it
Jasiobedzki et al. [24]	3D reconstruction	real-time reconstruction using stereo images, registered using ICP; resulting model is bent, authors plan to incorporate refraction to eliminate the error
Sedlazeck et al. [53]	3D reconstruction	classical, sequential SfM with adaptations to underwater environment; calibration below water, 2 coefficients for radial distortion, dense depth maps are used for model computation; additional color correction; absolute scale from navigation data
Pizarro et al. [46,47]	3D reconstruction	calibration below water including distortion; reconstructions based on 2 or 3 images are registered against each other by a graph based algorithm; Delaunay triangulation; usage of navigation data
Johnson et al. [25]	3D reconstruction	sparse sets of 3D points are meshed using a Delaunay triangulation and registered via SLAM utilizing navigation data; additional loop closing and color correction; can process thousands of images
Brandou et al. [3]	3D reconstruction	stereo rig is moved on predefined trajectory by a ROV arm; model is computed using dense depth maps; camera is calibrated on-site by deploying checkerboard on the sea floor
Negahdaripour et al. [40,41]	3D reconstruction	combination of optical and acoustic systems in one rig; calibration and reconstruction theory in presence of both: euclidean and spherical coordinates
Queiroz-Neto, Nascimento et al. [51,39]	underwater stereo	color correction routine is combined with stereo to match more robustly; no consideration of refraction
<b>Ray-based Models</b>		
Grossberg, Narasimhan et al. [17,38]	calibration, reconstruction	cameras are defined via their caustics; calibration routine using an active display; second paper specializes on underwater case with light sheet based reconstruction in small tank environments
Sturm, Chari et al. [55,54,7]	calibration, reconstruction	development of theory for generic cameras described only by their rays (assumption is that neighboring rays are close to each other); calibration by taking several checkerboard images; theory, but no implementation of ray-based cameras for refractive case (thin glass)
Wolff [62]	sea floor reconstruction	reconstruction of sea floor in simulator (small tank); ray-based, generic camera is approximated by several perspective cameras suitable for different image regions

Table 7. Paper overview

Authors	Application	Method
<b>Physical Underwater Models</b>		
Treibitz et al. [58]	calibration	physical model for underwater imaging is developed assuming thin glass and interface-sensor parallelism; analytical derivation of projection using cylinder coordinates; includes calibration routine; bridge to caustics
Telem et al. [56,57]	calibration	each point is mapped to a point eligible for perspective projection by moving the point in the image and computing the correct intersection with the optical axis
Kwon et al. [28,29]	calibration, measurement	refraction is modeled in combination with the DLT for pose estimation; assumed thin glass; no rotation between glass and camera sensor
Kunz et al. [27]	calibration	hemispherical and flat ports are modeled and synthetic data is used to experiment with inaccuracies using the perspective model; calibration routine is described, but not implemented; general case for non-parallel interfaces and thick glass
Li et al. [31,32]	calibration	development of complete physical model and its calibration; stereo rig is used to triangulate the derived rays in water; indexes of refraction are assumed to be known
Maas [33,34]	fluid flow measurement	a complete physical model for a rig is derived and implemented, then used for calibration; 3D points are moved to positions eligible for perspective projection by an iteration; rig is used to reconstruct fluid flow marked by suspended particles; parallelism is assumed; indexes of refraction cannot be calibrated
Putze [50,49]	calibration, fluid flow measurement	follow-up work to the above gaining more robustness
Förstner et al. [11]	reconstruction	[53] and its specialized epipolar geometry are used for surface reconstruction
Morris et al. [37]	wave surface reconstruction	a calibrated stereo rig views the bottom of a water tank on which a checkerboard pattern is placed; refraction is used to determine the wave's normals on the liquid's surface
Yamashita, Kawai et al. [64,65,26]	measurements	in small water tanks within a lab, a stereo system, a laser beam, or active patterns are used to gain reconstructions of objects completely or half emerged in the water
Ferreira et al. [10]	underwater stereo	the underwater model is linearized to compensate of the majority of the errors induced by using the perspective model for stereo
Chang et al. [6]	underwater SfM	underwater SfM with known vertical rotation (IMU) and explicit consideration of refraction

## 6 Conclusion and Future Work

We have discussed three different types of camera models, which are used to deal with refraction effects on underwater images.

First, it was shown that the often used pinhole camera model is invalid due to refraction at the camera housing, although it is common in the literature. The accuracy analysis for the perspective model shows that the model error is not negligible and grows with increasing interface distance and with stronger tilt of the interface with respect to the image sensor. Applications like stereo measurements, mosaicking for navigation, and Structure from Motion all rely on accurate geometrical measurements. Especially Structure from Motion is prone to errors due to drift in pose estimation and we believe that the systematic error caused by using a wrong model for refractive effects adds an unnecessary source of drift.

Second, the ray-based camera models have a completely derived theory for SfM, but no implementation has been tried on real underwater images yet. In addition, the high degree of freedom caused by individually parametrized rays for each pixel makes robust calibration difficult or even infeasible in underwater environments.

Third, physically modeled interfaces allow to compute refraction explicitly without needing a high degree of freedom. Only the parameters describing the underwater housing with respect to the camera are required in addition to the classic perspective camera model. Applications like SfM, mosaicking, and stereo based measurements could therefore profit from using such a model because the systematic error induced by using an approximate, perspective camera model can be eliminated by modeling refraction explicitly. Future Work will include robust calibration of the interface parameters and application of the physical model to underwater images.

## References

1. Aggarwal, M., Ahuja, N.: A pupil-centric model of image formation. *International Journal of Computer Vision* 48(3), 195–214 (2002)
2. Bouguet, J.Y.: Visual methods for three-dimensional modelling. Ph.D. thesis, California Institute of Technology Pasadena, CA, USA (1999), <http://etd.caltech.edu/etd/available/etd-02072008-115723/>
3. Brandou, V., Allais, A., Perrier, M., Malis, E., Rives, P., Sarrazin, J., Sarradin, P.: 3d reconstruction of natural underwater scenes using the stereovision system iris. In: *Proc. OCEANS 2007- Europe*, pp. 1–6 (2007)
4. Bryant, M., Wettergreen, D., Abdallah, S., Zelinsky, A.: Robust camera calibration for an autonomous underwater vehicle. In: *Australian Conference on Robotics and Automation (ACRA 2000)* (August 2000)
5. Carreras, M., Ridao, P., Garcia, R., Nicosevici, T.: Vision-based localization of an underwater robot in a structured environment. In: *Proceedings of the International Conference on Robotics and Automation, ICRA 2003*, vol. 1, pp. 971–976 (September 2003)



6. Chang, Y.J., Chen, T.: Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction. In: IEEE International Conference on Computer Vision, ICCV (2011)
7. Chari, V., Sturm, P.: Multiple-view geometry of the refractive plane. In: Proceedings of the 20th British Machine Vision Conference, London, UK (September 2009), <http://perception.inrialpes.fr/Publications/2009/CS09>
8. Costa, C., Loy, A., Cataudella, S., Davis, D., Scardi, M.: Extracting fish size using dual underwater cameras. *Aquacultural Engineering* 35(3), 218–227 (2006), <http://www.sciencedirect.com/science/article/B6T4C-4K4WMTT-1/2/d8103dd1d6946795645bf447642c7813>
9. Eustice, R., Singh, H., Howland, J.: Image registration underwater for fluid flow measurements and mosaicking. In: OCEANS 2000 MTS/IEEE Conference and Exhibition, vol. 3, pp. 1529–1534 (2000)
10. Ferreira, R., Costeira, J.P., Santos, J.A.: Stereo Reconstruction of a Submerged Scene. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) *IbPRIA 2005*. LNCS, vol. 3522, pp. 102–109. Springer, Heidelberg (2005)
11. Foerstner, W., Wolf, K.: Exploiting the multi view geometry for automatic surfaces reconstruction using feature based matching in multi media photogrammetry. In: Proceedings of the 19th ISPRS Congress, pp. 5B 900–907 (2000)
12. Fryer, J.G., Fraser, C.S.: On the calibration of underwater cameras. *The Photogrammetric Record* 12 (1986)
13. Garcia, R., Batlle, J., Cufi, X., Amat, J.: Positioning an underwater vehicle through image mosaicking. In: Proceedings of the International Conference on Robotics and Automation, ICRA 2001, vol. 3, pp. 2779–2784 (2001)
14. Glaeser, G., Schröcker, H.P.: Reflections on refractions. *Journal for Geometry and Graphics (JGG)* 4, 1–18 (2000)
15. Gracias, N., Santos Victor, J.: Underwater video mosaics as visual navigation maps. *Journal of Computer Vision and Image Understanding (CVIU)* 79(1), 66–91 (2000)
16. Gracias, N., van der Zwaan, S., Bernardino, A., Santos-Victor, J.: Mosaic-based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering* 28(4), 609–624 (2003)
17. Grossberg, M.D., Nayar, S.K.: The raxel imaging model and ray-based calibration. *International Journal of Computer Vision* 61(2), 119–137 (2005)
18. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press (2004), <http://www.amazon.com/Multiple-View-Geometry-Computer-Vision/dp/0521540518>
19. Harvey, E.S., Shortis, M.R.: Calibration stability of an underwater stereo-video system: Implications for measurement accuracy and precision. *Marine Technology Society Journal* 32, 3–17 (1998)
20. Hecht, E.: *Optik*. Oldenburg Verlag, Muenchen Wien (2005)
21. Heikkila, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 1106 (1997)
22. Hogue, A., German, A., Jenkin, M.: Underwater environment reconstruction using stereo and inertial data. In: IEEE International Conference on Systems, Man and Cybernetics, ISIC 2007, October 7–10, pp. 2372–2377 (2007)
23. Hogue, A., German, A., Zacher, J., Jenkin, M.: Underwater 3d mapping: Experiences and lessons learned. In: The 3rd Canadian Conference on Computer and Robot Vision, June 7–9, p. 24 (2006)

24. Jasiobedzki, P., Se, S., Bondy, M., Jakola, R.: Underwater 3d mapping and pose estimation for rov operations. In: OCEANS 2008, September 15-18, pp. 1–6 (2008)
25. Johnson-Roberson, M., Pizarro, O., Williams, S., Mahon, I.: Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics* 27 (2010)
26. Kawai, R., Yamashita, A., Kaneko, T.: Three-dimensional measurement of objects in water by using space encoding method. In: IEEE International Conference on Robotics and Automation, ICRA 2009, May 12-17, pp. 2830–2835 (2009)
27. Kunz, C., Singh, H.: Hemispherical refraction and camera calibration in underwater vision. In: OCEANS 2008, September 15-18, pp. 1–7 (2008)
28. Kwon, Y.: A camera calibration algorithm for the underwater motion analysis. In: 17th International Symposium on Biomechanics in Sports, ISBS - Conference Proceedings Archive (1999)
29. Kwon, Y., Casebolt, J.: Effects of light refraction on the accuracy of camera calibration and reconstruction in underwater motion analysis. *Sports Biomech.* 5(1), 95–120 (2006)
30. Lavest, J.-M., Rives, G., Lapresté, J.T.: Underwater Camera Calibration. In: Vernon, D. (ed.) ECCV 2000, Part II. LNCS, vol. 1843, pp. 654–668. Springer, Heidelberg (2000)
31. Li, R., Tao, C., Zou, W.: An underwater digital photogrammetric system for fishery geomatics. In: Intl. Archives of PRS, vol. XXXI, pp. 319–323 (1996)
32. Li, R., Li, H., Zou, W., Smith, R., Curran, T.: Quantitative photogrammetric analysis of digital underwater video imagery. *IEEE Journal of Oceanic Engineering* 22(2), 364–375 (1997)
33. Maas, H.G.: Digitale Photogrammetrie in der dreidimensionalen Stroemungsmesstechnik. Ph.D. thesis, Eidgenoessische Technische Hochschule Zuerich (1992)
34. Maas, H.G.: New developments in multimedia photogrammetry. In: Optical 3-D Measurement Techniques III. Wichmann Verlag, Karlsruhe (1995)
35. McGlone, J.C. (ed.): Manual of Photogrammetry, 5th edn. ASPRS (2004), <http://www.amazon.de/Manual-Photogrammetry-American-Society/dp/0937294012>
36. Mobley, C.D.: Light and Water: Radiative Transfer in Natural Waters. Academic Press (1994)
37. Morris, N., Kutulakos, K.N.: Dynamic refraction stereo. In: Proc. 10th Int. Conf. Computer Vision, pp. 1573–1580 (2005)
38. Narasimhan, S.G., Nayar, S.: Structured light methods for underwater imaging: light stripe scanning and photometric stereo. In: Proceedings of 2005 MTS/IEEE OCEANS, vol. 3, pp. 2610–2617 (September 2005)
39. Nascimento, E.R., Campos, M.F.M., Barros, W.F.: Stereo based structure recovery of underwater scenes from automatically restored images. In: Nonato, L.G., Scharcanski, J. (eds.) Proceedings SIBGRAPI 2009 (Brazilian Symposium on Computer Graphics and Image Processing), October 11-14. IEEE Computer Society, Los Alamitos (2009), <http://urlib.net/sid.inpe.br/sibgrapi@80/2009/08.18.16.07>
40. Negahdaripour, S., Sekkati, H., Pirsiavash, H.: Opti-acoustic stereo imaging, system calibration and 3-d reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, June 17-22, pp. 1–8 (2007)
41. Negahdaripour, S., Sekkati, H., Pirsiavash, H.: Opti-acoustic stereo imaging: On system calibration and 3-d target reconstruction. *IEEE Transactions on Image Processing* 18(6), 1203–1214 (2009)

42. Negahdaripour, S., Xu, X., Khamene, A., Awan, Z.: 3-d motion and depth estimation from sea-floor images for mosaic-based station-keeping and navigation of rovs/auvs and high-resolution sea-floor mapping. In: Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles, AUV 1998, pp. 191–200 (August 1998)
43. Pessel, N., Opderbecke, J., Aldon, M.J.: Camera self-calibration in underwater environment. In: WSCG (2003)
44. Pessel, N.: Auto-Calibrage d'une Caméra en Milieu Sous-Marin. Ph.D. thesis, Université Montpellier II (2003)
45. Pessel, N., Opderbecke, J., Aldon, M.J.: An experimental study of a robust self-calibration method for a single camera. In: 3rd International Symposium on Image and Signal Processing and Analysis, ISPA 2003. Sponsored by IEEE and EURASIP, Rome, Italie (September 2003)
46. Pizarro, O., Eustice, R., Singh, H.: Relative pose estimation for instrumented, calibrated imaging platforms. In: DICTA, pp. 601–612 (2003)
47. Pizarro, O., Eustice, R., Singh, H.: Large area 3d reconstructions from underwater surveys. In: Proc. OCEANS 2004, MTTs/IEEE TECHNO-OCEAN 2004, vol. 2, pp. 678–687 (2004)
48. Press, W.H., Vetterling, W.T., Teukolsky, S.A., Flannery, B.P.: Numerical Recipes in C++: the art of scientific computing, 2nd edn. Cambridge University Press, New York (2002)
49. Putze, T.: Erweiterte verfahren zur mehrmedienphotogrammetrie komplexer körper. In: Beiträge der Oldenburger 3D-Tage 2008. Herbert Wichmann Verlag, Heidelberg (2008)
50. Putze, T.: Geometrische und stochastische Modelle zur Optimierung der Leistungsfähigkeit des Stroemungsmessverfahrens 3D-PTV. Ph.D. thesis, Technische Universität Dresden (2008)
51. Queiroz-Neto, J.P., Carceroni, R., Barros, W., Campos, M.: Underwater stereo. In: Proc. 17th Brazilian Symposium on Computer Graphics and Image Processing, October 17-20, pp. 170–177 (2004)
52. Schiller, I., Beder, C., Koch, R.: Calibration of a pmd camera using a planar calibration object together with a multi-camera setup. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXI. ISPRS Congress, Beijing, China, vol. XXXVII, Part B3a, pp. 297–302 (2008), [www.mip.informatik.uni-kiel.de/tiki-index.php?page=Ingo+Schiller-23k-](http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Ingo+Schiller-23k-)
53. Sedlazeck, A., Koser, K., Koch, R.: 3d reconstruction based on underwater video from rov kiel 6000 considering underwater imaging conditions. In: Proc. OCEANS 2009, OCEANS 2009-EUROPE, May 11-14, pp. 1–10 (2009)
54. Sturm, P., Ramalingam, S., Lodha, S.: On calibration, structure from motion and multi-view geometry for generic camera models. In: Daniilidis, K., Klette, R. (eds.) Imaging Beyond the Pinhole Camera, Computational Imaging and Vision, vol. 33. Springer (August 2006), <http://perception.inrialpes.fr/Publications/2006/SRL06>
55. Sturm, P., Ramalingam, S.: A Generic Concept for Camera Calibration. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004, Part II. LNCS, vol. 3022, pp. 1–13. Springer, Heidelberg (2004)
56. Telem, G., Filin, S.: Calibration of consumer cameras in a multimedia environment. In: ASPERS 2006 Annual Conference (2006)

57. Telem, G., Filin, S.: Photogrammetric modeling of underwater environments. *ISPRS Journal of Photogrammetry and Remote Sensing* 65(5), 433–444 (2010), <http://www.sciencedirect.com/science/article/B6VF4-50F9H66-1/2/d8dba566f79b0a207e13a6aa2bf3f69d>
58. Treibitz, T., Schechner, Y., Singh, H.: Flat refractive geometry. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pp. 1–8 (2008)
59. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle Adjustment – A Modern Synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) *Vision Algorithms 1999*. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
60. Trucco, E., Doull, A., Odone, F., Fusiello, A., Lane, D.: Dynamic video mosaicing and augmented reality for subsea inspection and monitoring. In: *Oceanology International, United Kingdom* (2000)
61. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, an efficient and accurate camera calibration technique. *IEEE Journal of Robotics and Automation* RA-3(4), 323–344 (1987)
62. Wolff, K.: Zur Approximation allgemeiner optischer Abbildungsmodelle und deren Anwendung auf eine geometrisch basierte Mehrbildzuordnung am Beispiel einer Mehrmedienabbildung. Ph.D. thesis, Rheinische Friedrich-Wilhelms-Universitaet Bonn (2007)
63. Xu, X., Negahdaripour, S.: Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwatervehicle. In: *ICRA 2001*, pp. 2759–2766 (2001)
64. Yamashita, A., Hayashimoto, E., Kaneko, T., Kawata, Y.: 3-d measurement of objects in a cylindrical glass water tank with a laser range finder. In: *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, October 27–31, vol. 2, pp. 1578–1583 (2003)
65. Yamashita, A., Fujii, A., Kaneko, T.: Three dimensional measurement of objects in liquid and estimation of refractive index of liquid by using images of water surface with a stereo vision system. In: *ICRA*, pp. 974–979 (2008)
66. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: *Proceedings of the International Conference on Computer Vision, Corfu, Greece*, pp. 666–673 (1999), <http://www.citeulike.org/user/snsinha/article/238276>

# An Introduction to Random Forests for Multi-class Object Detection

Juergen Gall<sup>1,2,\*</sup>, Nima Razavi<sup>1</sup>, and Luc Van Gool<sup>1,3</sup>

<sup>1</sup> Computer Vision Laboratory, ETH Zurich  
{gall,nrazavi,vangool}@vision.ee.ethz.ch

<sup>2</sup> Max Planck Institute for Intelligent Systems

<sup>3</sup> ESAT/IBBT, Katholieke Universiteit Leuven

**Abstract.** Object detection in large-scale real-world scenes requires efficient multi-class detection approaches. Random forests have been shown to handle large training datasets and many classes for object detection efficiently. The most prominent example is the commercial application of random forests for gaming [37]. In this paper, we describe the general framework of random forests for multi-class object detection in images and give an overview of recent developments and implementation details that are relevant for practitioners.

**Keywords:** multi-class object detection, Hough forest, regression forest, random forest.

## 1 Introduction

Object detection for real-world applications is still a challenging problem. While recent research datasets like PASCAL VOC [12], ImageNet [10], or the Caltech Pedestrian Dataset [11] increase the amount of training and testing examples to get closer to real-world problems, the ability of detectors to process large data sets in reasonable time becomes another important issue besides accuracy. It is not only the number of training examples that matters, but also the number of classes.

A family of methods that can handle large amount of training data efficiently and that are inherently suited for multi-class problems are based on random forests [15]. Random forests are ensembles of randomized decision trees that can be applied for regression [8,13,19], classification tasks [26,28,30,6,40,4,35,38,37], and even both at the same time [16,31,39,18,14]. The most prominent application of random forest is the detection of human body parts from depth data [37]. The method was trained on 900k training examples to detect 31 body parts (classes) and runs at around 200 frames per second on the Xbox GPU. This commercial application demonstrates the practicability of random forests for large-scale real-world computer vision problems.

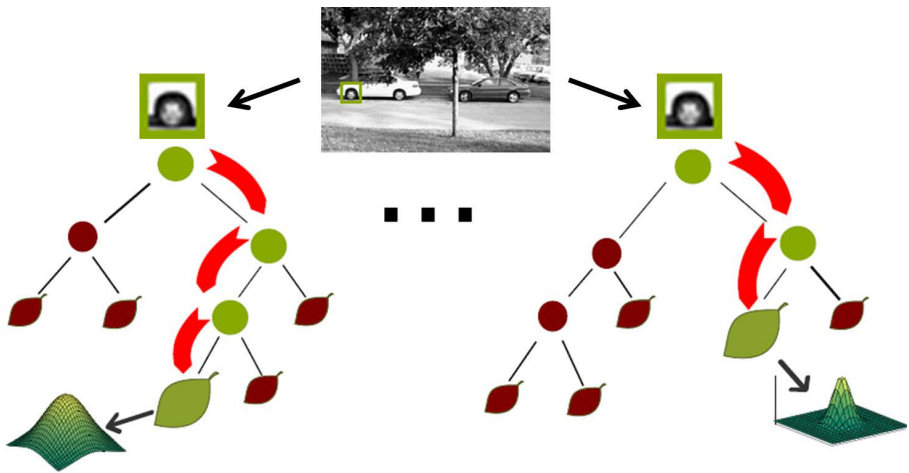
---

\* This work has been partially funded by the EU projects IURO (FP7-ICT-248314) and RADHAR (FP7-ICT-248873). The paper contains content that has been previously published in [18,32,33].

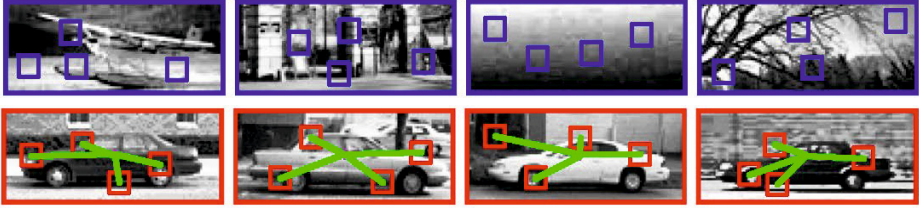
The scope of this paper is to give an introduction to random forests in the context of multi-class object detection and to give an overview of recent developments. For a more general discussion on random forests, we refer to the book [5] and the tutorial [7]. Rather than providing a detailed experimental evaluation which has been already presented in the referenced works, the paper serves more as a guide for practitioners.

## 2 Random Forests for Object Detection

A random forest consists of a set of trees  $T_t$  where each tree consists of split nodes and leaves as illustrated in Figure 1. The split nodes evaluate each arriving image patch and, depending on the appearance of the patch, pass it to the left or right child. Each leaf  $L$  stores the statistics of the image patches that arrived during training. For a classification task, it is the probability for each class  $c$ , denoted by  $p(c|L)$ . For a regression task, it is a distribution over the continuous parameter  $x \in \mathbb{R}^H$  that one wants to estimate. While image segmentation is a typical classification task where one wants to estimate the class label for each image patch, object localization can be regarded as a regression problem where each patch of the object predicts the location of the object in the image. Since object detection involves both classifying patches belonging to an object and using them to regress the location and scale of the object, random forests for object detection need to be trained to satisfy both objectives.



**Fig. 1.** A random forest consists of a set of trees that map an image patch to a distribution stored at each leaf. The disks indicate split nodes that evaluate the appearance of a patch and pass it to the right or left child until a leaf is reached.



**Fig. 2.** For training, a subset of image patches is taken from the entire training set. In the simplest case, there are only two classes; one containing negative or background examples (*blue*) and another containing positive examples (*red*). While the class labels are required to distinguish object patches from background patches (*classification*), additional offset vectors of the positive patches to the center of the object are stored (*green*). The offset vectors will be used to predict the location of the object (*regression*).

## 2.1 Training

For training, a set of images is collected where each object is annotated by a bounding box and the class label  $c$ . The background images are only annotated by the class label. In order to handle large amount of training data and to avoid overfitting, randomness is introduced by training each tree on a randomly sampled subset of the training data [5]. For object detection, this means to randomly select a subset of training images for each class. From the selected images, only a subset of image patches is then sampled and used for training as illustrated in Figure 2. For each sampled patch  $\mathcal{P}_i$  that does not belong to the background, the offset to a reference point of the object  $\mathbf{d}_i$  is stored. Ideally, the reference point is always the same for all training instances of a class, e.g., the head of a pedestrian. However, taking the center of the bounding box as reference point is usually a more practical choice. In general, the reference point does not need to be the center of the object, but it should be as consistent as possible among training examples. Scale is handled during testing and the positive examples are scaled to a unit size  $s_u$ . A good choice for object detection has been to use image patches of size  $16 \times 16$  pixels and scale the images such that the longest spatial dimension of the bounding box is about 100 pixels [16]. In this setting, a patch covers meaningful parts like a wheel of a car or the head of a human as shown in Figures 2 and 4. In case of tight bounding boxes around the objects, it is beneficial to consider all patches for sampling that have the patch center inside of a bounding box. In this way, important boundary information can be better captured [9].

In summary, we have a set of training patches  $\{\mathcal{P}_i = (\mathcal{I}_i, c_i, \mathbf{d}_i)\}$  that are randomly sampled from the examples where,

- $\mathcal{I}_i$  are the extracted image features of the patch,
- $c_i$  is the class label for the exemplar, the patch is sampled from,
- $\mathbf{d}_i$  is a offset vector from the patch center to the reference point.

Patches sampled from background images have a pseudo offset, i.e.,  $\mathbf{d}_i = 0$ . We denote the set of randomly sampled training patches for a tree  $T_t$  by  $A = \{\mathcal{P}_i\}$ .

In order to train a tree that can be used for object detection, one has to find a split function

$$f_\phi(\mathcal{P}) \in \{0, 1\} \quad (1)$$

for each non-leaf node that separates the training patches in an optimal way. The split functions are therefore also termed as weak learners [7]. The split function evaluates one or more image features of the patch  $\mathcal{P}$  and sends it to the left ( $f_\phi(\mathcal{P}) = 0$ ) or right child ( $f_\phi(\mathcal{P}) = 1$ ) of the node; see Figure 1. The split functions are parametrized by a set of parameters  $\phi$  that need to be optimized during training.

Each tree can be trained in parallel using the general random forest framework [5]. Starting at the root node with the training set  $A_{node} = A$ , a tree grows recursively:

1. Generate a random set of parameters  $\Phi = \{\phi_k\}$ .
2. Divide the set of patches  $A_{node}$  into two subset  $A_L$  and  $A_R$  for each  $\phi \in \Phi$ :

$$A_L(\phi) = \{\mathcal{P} \in A_{node} | f_\phi(\mathcal{P}) = 0\} \quad (2)$$

$$A_R(\phi) = \{\mathcal{P} \in A_{node} | f_\phi(\mathcal{P}) = 1\} \quad (3)$$

3. Select the split parameters  $\phi^*$  that maximize a gain function  $g$ :

$$\phi^* = \operatorname{argmax}_{\phi \in \Phi} g(\phi, A_{node}) \quad (4)$$

where

$$g(\phi, A_{node}) = \mathcal{H}(A_{node}) - \sum_{S \in \{L, R\}} \frac{|A_S(\phi)|}{|A_{node}|} \mathcal{H}(A_S(\phi)). \quad (5)$$

Depending on the task,  $\mathcal{H}(A)$  is chosen such that  $g$  measures the gain of the classification or regression performance of the children in comparison to the current node.

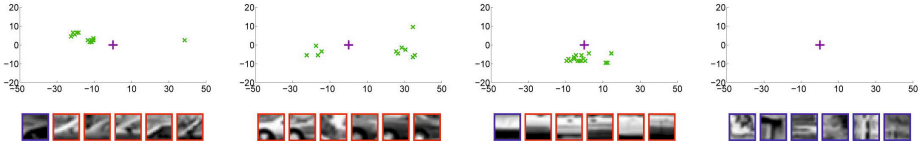
4. Continue growing with the training subsets  $A_L$  and  $A_R$  if some predefined stopping criteria are not satisfied; otherwise, create a leaf node and store the statistics of the training data  $A_{node}$ .

Step 1 is another source of randomness that reduces training time whereas evaluating all parameters  $\phi$  would be infeasible in many cases. While the family of split functions  $f_\phi$ , the measure  $\mathcal{H}$ , and the stopping criteria will be discussed in Section 3, we continue with the prediction model stored at each leaf  $L$ .

In the context of object detection, we are interested in the class probability and the spatial distribution of the training patches for each class. The class probability  $p(c|L)$  can be estimated by

$$p(c|L) = \frac{|A_c^L| \cdot r_c}{\sum_c (|A_c^L| \cdot r_c)}; \quad r_c = \frac{|A|}{|A_c|} \quad (6)$$





**Fig. 3.** Visualization of some leaves of a tree for detecting cars (side-view; two classes). Each leaf node  $L$  stores the probability of a patch belonging to the object class  $p(c|L)$ , estimated by the proportion of patches from the positive (red) and negative examples (blue) reaching the leaf during training. For the positive class, the offset vectors  $\mathbf{d} \in D_c^L$  are shown (green). The underlying distribution  $p(\mathbf{d}|c, L)$  is multimodal. The positive training examples falling inside each of the first three leaves can be associated with different parts of a car. The last leaf contains only negative patches. The image has been taken from [18].

where  $A^L$  is the set of training patches reaching the leaf  $L$  after training,  $A$  the entire training set used for training the tree, and  $A_c$  the patches in  $A$  with class label  $c$ . The factor  $r_c$  compensates for the sample bias that might have been introduced when the number of training examples is not well distributed among classes. The spatial distribution for each class,  $p(\mathbf{d}|c, L)$ , is obtained by estimating the continuous distribution from the offset samples  $\mathbf{d} \in D_c^L$  of the patches  $A_c^L$ . While more details will be given in Section 3, the statistics of a few example leaves are shown in Figure 3.

## 2.2 Detection

For detecting an object, image patches are sampled from a test image and passed through the trees as shown in Figure 1. The image patches can be densely sampled or subsampled as for training. Each patch  $\mathcal{P}(\mathbf{y})$  sampled from image location  $\mathbf{y}$  ends in a leaf  $L_t(\mathbf{y})$  for each tree  $T_t$ . In order to locate an object in the image, we evaluate the probability of an object hypothesis  $\mathbf{h}(c, \mathbf{x}, s)$ , i.e., the probability of an object belonging to class  $c$  with size  $s$  and its reference point at  $\mathbf{x}$ . Besides of scale, additional parameters of the object like depth [39], viewpoint [32], or aspect ratio [16] can be estimated.

The probability  $p(\mathbf{h}|L_t(\mathbf{y}))$  for a single patch and a single tree is then given by

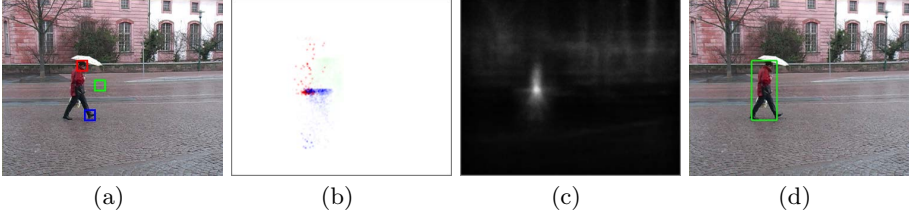
$$p(\mathbf{h}(c, \mathbf{x}, s)|L_t(\mathbf{y})) = p(\mathbf{d}(\mathbf{x}, \mathbf{y}, s)|c, L_t(\mathbf{y})) p(c|L_t(\mathbf{y})) \quad (7)$$

where

$$\mathbf{d}(\mathbf{x}, \mathbf{y}, s) = \frac{s_u(\mathbf{y} - \mathbf{x})}{s}. \quad (8)$$

The term  $\mathbf{d}(\mathbf{x}, \mathbf{y}, s)$  is basically the offset between  $\mathbf{y}$  and  $\mathbf{x}$  given the hypothesis size  $s$ . Note that the unit size  $s_u$  and the two probabilities  $p(\mathbf{d}|c, L_t(\mathbf{y}))$  and  $p(c|L_t(\mathbf{y}))$ , cf. Equation (6), are known from training as explained in Section 2.1. The derivation of Equation (7) is straightforward and given in [18]. While random

<sup>1</sup> We abbreviate  $\mathbf{h}(c, \mathbf{x}, s)$  to  $\mathbf{h}$  and  $\mathbf{d}(\mathbf{x}, \mathbf{y}, s)$  to  $\mathbf{d}$ .



**Fig. 4.** For each of the three patches emphasized in (a), the random forest trained on pedestrians casts weighted votes about the possible location of a pedestrian (b) (each color channel corresponds to the vote of a sample patch). Note the weakness of the vote from the background patch (green). After the votes from all patches are aggregated (c) (white corresponds to a high value), the pedestrian can be detected (d) by searching the mode of (c). The images have been taken from [18].

regression forests or classification forests model only one of the two terms in Equation (7), the distribution  $p(\mathbf{h}|L_t(\mathbf{y}))$  combines both the regression and the classification objective.

The distribution  $p(\mathbf{d}|c, L_t(\mathbf{y}))$  can be modeled by a set of votes  $\mathbf{d} \in D_c^{L_t(\mathbf{y})}$ . In this case, Equation (7) becomes

$$p(\mathbf{h}(c, \mathbf{x}, s)|L_t(\mathbf{y})) = \frac{1}{|D_c^{L_t(\mathbf{y})}|} \left( \sum_{\mathbf{d} \in D_c^{L_t(\mathbf{y})}} \delta_{\mathbf{d}} \left( \frac{s_u(\mathbf{y} - \mathbf{x})}{s} \right) \right) p(c|L_t(\mathbf{y})), \quad (9)$$

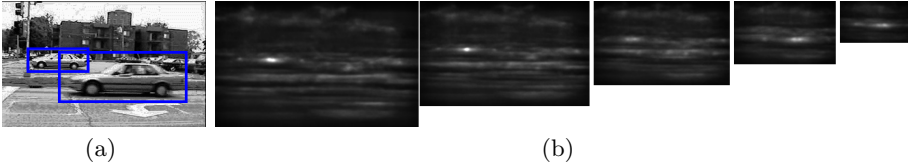
where  $\delta$  is a Dirac measure. Since the distribution can be regarded as weighted votes  $\mathbf{d} \in D_c^{L_t(\mathbf{y})}$  to be cast into a Hough space, regression trees are also termed Hough forests [16] in the context of object detection. Figures 4 (a) and (b) show the votes or distribution of three patches. While the head (red) patch yields a distribution with one strong mode, the patch of the right feet (blue) is similar to the left feet in appearance and thus yielding a distribution with two modes. The impact of the class probability can be observed for the background patch (green). Since the probability of this patch belonging to the object is close to zero, the votes are barely visible.

While Equation (7) models the probability for a single tree, the probabilities of all trees are averaged, i.e.,

$$p(\mathbf{h}(c, \mathbf{x}, s)|\mathcal{P}(\mathbf{y})) = \frac{1}{|\{T_t\}|} \sum_t p(\mathbf{h}(c, \mathbf{x}, s)|L_t(\mathbf{y})). \quad (10)$$

Alternatively, the probabilities can also be multiplied but averaging is more efficient [7]. Similarly, the distributions over all image patches can be either accumulated as in [18]:

$$p(\mathbf{h}(c, \mathbf{x}, s)|\mathcal{I}) = \frac{1}{|\Omega|} \sum_{\mathbf{y} \in \Omega} p(\mathbf{h}(c, \mathbf{x}, s)|\mathcal{P}(\mathbf{y})); \quad (11)$$



**Fig. 5.** In order to detect objects at different scales (a), the original image is scaled by the inverse expected sizes (b). The modes are detected in the joint space of image location and scale (white corresponds to a high value). The small car yields a peak on the two left images and the large car yields a peak on the right images.

or multiplied as in [2]. An example using Equation (11) for a single scale is shown in Figure 4 (c). Multiple scales can be handled by processing the image at different scales as shown in Figure 5. In order to detect an object of size  $s$ , giving the training size  $s_u$ , the image is scaled by  $\frac{s_u}{s}$ . In this way, the scale factor in Equation (8) is already taken into account. Object detection can then be performed by using mean shift to detect the modes of Equation (11); see Figures 4 and 5.

### 3 Implementation Details

So far, the general framework has been described. In this section, we discuss variations and implementation details that are relevant for applications.

#### 3.1 Features and Binary Tests

Actually any kind of image feature can be used that is useful for object detection. This includes sparse features like SIFT [27] or SURF [3], but usually one relies on low-level features like color, gradients, or Gabor filters that can be efficiently computed. In contrast to manual designed feature descriptors, the random forest selects a split function (11) at each non-leaf node during training. All patches ending in one leaf are therefore described by the split functions from the root to the leaf. The split functions, however, can be directly optimized for the task of object detection.

A set of features obtained from simple pixel tests using intensity and first-order gradients are shown in Figure 3. The used pixel tests are defined by:

$$f_\phi(\mathcal{P}) = \begin{cases} 0 & \text{if } I^f(\mathbf{p}) - I^f(\mathbf{q}) < \tau \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

where the parameters  $\phi = \{\mathbf{p}, \mathbf{q}, f, \tau\}$  comprise two pixel locations within the patch, a low-level image feature  $I^f$  of the patch, and a threshold  $\tau$ . The pixel differences introduce invariance with respect to a constant change of the image features  $I^f$ . In [28], only a pixel value is thresholded, i.e.,  $I^f(\mathbf{p}) < \tau$ . More general tests than (12) have been used in [8,13], where the feature values over two regions  $Q$  and  $P$  are averaged:

$$f_\phi(\mathcal{P}) = \begin{cases} 0 & \text{if } \frac{1}{|P|} \sum_{\mathbf{p} \in P} I^f(\mathbf{p}) - \frac{1}{|Q|} \sum_{\mathbf{q} \in Q} I^f(\mathbf{q}) < \tau \\ 1 & \text{otherwise.} \end{cases} \quad (13)$$

The regions are rectangles within the patch such that the average can be efficiently computed by integral images. In general, several features can be also combined for a single test:

$$f_\phi(\mathcal{P}) = \begin{cases} 0 & \text{if } \tau_1 < \sum_f w_f \left( \frac{1}{|P_f|} \sum_{\mathbf{p} \in P_f} I^f(\mathbf{p}) \right) < \tau_2 \\ 1 & \text{otherwise,} \end{cases} \quad (14)$$

where  $w_f$  is a weight between the features, e.g.,  $w_f \in \{-1, 0, 1\}$ , and  $\tau_1$  is an additional threshold [7]. While more complex split functions allow a better separation at each node, they also involve more parameters to estimate and increase the chance of overfitting [7]. Even for Equation (12) the patch size has an impact on the detection performance although it is not very sensitive to the exact size [18].

In practice, split tests of type (12) or (13) have shown to give a good performance for object detection. While in case of depth data, using only depth data already gives good results [19,37], 32 image features have been used in [18] for object detection. Similarly to the number of parameters  $\phi$ , the number of image features can result in overfitting, i.e., a random forest trained with less image features might perform better than a forest with many features.

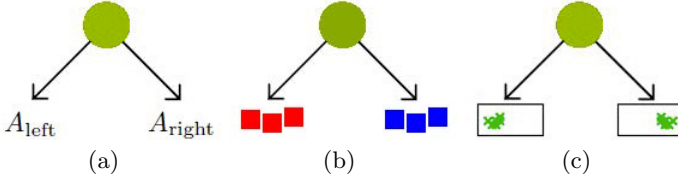
To increase the robustness of features, one can introduce some variance to the patches by transforming them. For instance, patches of various sizes and orientations are used in the context of classification [28]. In [24], additional patches are tracked in video clips and added to the training data as temporal pairs. When measuring the goodness of a split, one can enforce that temporal pairs are not split. In this way, one can introduce some robustness of the features with respect to small appearance changes over time. Measuring the goodness of a split will be discussed in the next section.

In order to make the evaluation of the random set of parameters  $\Phi$  more efficient, one generates the parameters of a split function  $\phi$  without a threshold  $\tau$ . The real-valued function  $f_{\phi \setminus \tau}$  is then applied to all patches  $\mathcal{P}_i \in A$ , which are sorted such that  $f_{\phi \setminus \tau}(\mathcal{P}_i) \leq f_{\phi \setminus \tau}(\mathcal{P}_j)$  for  $i \leq j$ . In this way, many thresholds can be efficiently evaluated for the split function  $f_\phi$ . In [18], 10 randomly generated thresholds are generated for each of the 2000 functions  $f_{\phi \setminus \tau}$ , yielding 20k split functions for  $\Phi$  in total.

### 3.2 Goodness of Split

Having defined a family of split functions, one has to measure the quality of a split (5) by defining  $\mathcal{H}(A)$ . Depending on the task, one can define a classification or regression functional; see Figure 6. An entropy-based classification functional can be computed by

$$\mathcal{H}_c(A) = - \sum_c p(c|A) \log(p(c|A)), \quad (15)$$



**Fig. 6.** (a) Each split function separates the training data at a node. (b) The classification objective aims to separate patches with different class labels. (c) The regression objective aims to maximize the localization accuracy of the offsets.

where  $p(c|A)$  is computed as in Equation (6). The functional tends to separate patches with different class labels in order to get leaves with low uncertainty for  $p(c|L)$ .

As in [7], one can also define a regression functional in a similar way by:

$$\mathcal{H}_r(A) = - \sum_c \frac{1}{|A|} \sum_{\mathcal{P} \in A} \int_{\mathbf{d}} p(\mathbf{d}|c, \mathcal{P}) \log(p(\mathbf{d}|c, \mathcal{P})) \, d\mathbf{d}, \tag{16}$$

to obtain leaves with a low uncertainty for  $p(\mathbf{d}|c, L)$ . While  $\mathcal{H}_r(A)$  can be efficiently computed under the assumption that  $p(\mathbf{d}|c, \mathcal{P})$  are Gaussian distributions, the functional becomes too expensive for more general distributions. In [18], a functional that is more efficient to compute has been used for regression:

$$\mathcal{H}_r(A) = \sum_c \left( \sum_{\mathbf{d} \in D_c^A} \left\| \mathbf{d} - \frac{1}{|D_c^A|} \sum_{\mathbf{d}' \in D_c^A} \mathbf{d}' \right\|^2 \right). \tag{17}$$

Using functional (16) with a Gaussian assumption or functional (17) is not optimal since both functionals assume a unimodal distribution of the offsets, which is not correct for object detection as illustrated in Figure 3. In practice, these approximations are, however, preferred due to training efficiency.

For object detection, one is interested in minimizing the uncertainties for  $p(c|L)$  and  $p(\mathbf{d}|c, L)$ . Therefore, one searches for a split function  $f_\phi$  that maximizes the gain (5) using  $\mathcal{H}_c$  and  $\mathcal{H}_r$ , denoted by  $g_c(\phi, A)$  and  $g_r(\phi, A)$ , respectively. While the objective is randomly selected at each node in [18], [31] uses a weighted combination of  $g_c(\phi, A)$  and  $g_r(\phi, A)$ :

$$g_{cr}(\phi, A) = g_c(\phi, A) + w(A)g_r(\phi, A). \tag{18}$$

In [31],  $w(A)$  is only defined for a two class problem with a positive and a negative class:

$$w(A) = \alpha \max(p(c_{\text{pos}}|A) - t_p, 0). \tag{19}$$

In general, the measure  $g_{cr}(\phi, A)$  tries to separate patches with different class labels first. If the purity of positive patches exceeds a given threshold  $t_p$ , the impact of the regression functional  $g_r(\phi, A)$ , weighted by the constant  $\alpha$ , increases.

In [14], several weights for combining  $g_c$  and  $g_r$  based on the depth of the node and including (19) have been evaluated in the context of head pose estimation. They showed that a random approach as in [18] gives very similar performance to weighting schemes with optimized parameters. The random approach, however, does not require additional parameters and is more efficient since only one functional needs to be evaluated at each node.

While combining the classification term with the regression term improved the performance for object detection in [18], the classification term gave the best performance in the context of body part detection [19]. Body part detection is a special case since all classes are spatially connected and it seems that enforcing a local separation based on body part labels seems to be more appropriate than making a unimodal approximation of the spatial distributions.

To avoid overfitting, the parameters of the split functions can be regularized. For instance, the weights  $w_f$  of the split functions (14) can be regularized by  $g(\phi, A) - \lambda \sum_f \|w_f\|^2$  [29]. In [24], pairs of patches ( $\mathcal{P}^1, \mathcal{P}^2$ ) are used for regularization:

$$-\lambda \left( \frac{1}{|B_{pos}|} \sum_{B_{pos}} \mathbb{I}(f_\phi(\mathcal{P}^1) \neq f_\phi(\mathcal{P}^2)) + \frac{1}{|B_{neg}|} \sum_{B_{neg}} \mathbb{I}(f_\phi(\mathcal{P}^1) = f_\phi(\mathcal{P}^2)) \right), \quad (20)$$

where  $\mathbb{I}$  is an indicator function. The regularizer enforces that patches that are similar under certain transformations, i.e.,  $(\mathcal{P}^1, \mathcal{P}^2) \in B_{pos}$ , are not separated while patches that are dissimilar,  $B_{neg}$ , are separated. The regularizer can be used to introduce some robustness of the features with respect to specific transformations. In contrast to the training data  $A$ , the pairs in  $B$  do not contain class labels and can be collected from other sources. For instance, tracked patches in arbitrary video sequences were used in [24] to build pairs for regularization in the context of object detection and tracking.

Equation (20) relates to semi-supervised learning that can be implemented in an iterative approach as in [25] or by computing the unsupervised gain  $g_u$  that prefers to cluster patches of similar appearance [7]. Since the unsupervised gain does not depend on labels, it can be computed over the union of the labeled set  $A$  and an additional unlabeled set  $B$ . The supervised and unsupervised gain can be combined by:

$$g(\phi, A) + \lambda g_u(\phi, A \cup B) \quad (21)$$

where  $g_u$  is defined by using

$$\mathcal{H}_u(A \cup B) = - \int_{\mathcal{I}} p(\mathcal{I}|A \cup B) \log(p(\mathcal{I}|A \cup B)) \, d\mathcal{I} \quad (22)$$

in Equation (5). As Equation (16), the term  $\mathcal{H}_u$  can be efficiently computed under the assumption that the appearance of the patches of the set  $A \cup B$  can be approximated by a Gaussian distribution  $p(\mathcal{I}|A \cup B)$ ; otherwise the evaluation becomes too expensive.

Regularizers and semi-supervised learning are important when the set of labeled training data is rather small to avoid overfitting. In case of large amount of labeled

training data, the set of patches does not fit in the memory and on-line learning [36] or subsampling strategies [36,19] can be used. These strategies can be easily implemented using only a subset of the training data  $A' \subset A$  for training a tree until a certain size. For the next step, another subset  $A'' \subset A$  is sampled and passed through the previously learned tree. The training is then continued until the tree has reached a final size. After the parameters of the split functions at the non-leaf nodes have been optimized, the distributions at the leaves can be computed from the full training set  $A$  by passing all patches through the tree and updating the offsets  $D_c^L$  and the histograms of the class labels  $|A_c^L|$  at the leaves. On-line learning or updating the leaf statistics is also performed for object tracking [17,36,20] where the training examples arrive sequentially over time.

### 3.3 Stopping Criteria

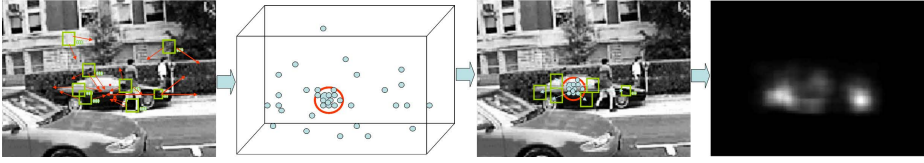
There are three main criteria for stopping the growing of a tree. The maximum depth of a tree, a minimum number of samples arriving at a node during training  $|A_{node}|$ , and a threshold based on the gain measure  $g(\phi^*, A_{node})$  (5). While the gain should be always strictly positive, i.e.,  $g(\phi^*, A_{node}) > 0$ , finding a good threshold is difficult. Therefore, limiting the tree depth and the minimum number of samples are more practical criteria. In the context of on-line learning [36], it has been shown that  $|A_{node}| > \epsilon$  is a sufficient criteria and an additional thresholding of the gain is not necessary. The optimal depth, however, depends on the amount of training data. For instance, the optimal performance for detecting organs in CT scans has been achieved by training 12 trees with depth 7 on the available 55 training examples [8]. In [19], 3 trees with depth 20 trained on 300k training examples performed well. While the number of trees is less critical since the performance does not decrease with more trees, trees that are too deep can have a negative impact on the performance due to overfitting [8].

### 3.4 Leaf Prediction Model

While  $p(c|L)$  is defined in (6), there are several choices for modeling the spatial distributions  $p(\mathbf{d}|c, L)$  at the leaf  $L$ . In [16], a Parzen estimate with a Gaussian kernel  $K$  is used to reconstruct the distribution from the samples:

$$p(\mathbf{d}|c, L) = \frac{1}{|D_c^L|} \left( \sum_{\mathbf{d}' \in D_c^L} K(\mathbf{d}' - \mathbf{d}) \right). \quad (23)$$

Although the non-parametric approach is very general, it does not scale with the number of offsets per class  $|D_c^L|$ . For many training examples, it is therefore recommended to approximate the distributions by a Gaussian mixture model as in [19,21]. Since in both cases a multimodal regression functional (16) is too expensive to evaluate for training, it is therefore approximated by a more simple, unimodal measure. In case of pose estimation [8,13],  $p(\mathbf{d}|c, L)$  is even approximated by a single Gaussian. Although this makes the testing very efficient, it is not an appropriate choice for object detection as indicated by the leaf distributions shown in Figure 3.



**Fig. 7.** Object detection with backprojection. From left to right: After passing the patches of the test image through the trees, the votes are collected. The mode of the distribution the votes are sampled from is detected by mean shift. The votes are backprojected to the image showing the image patches that voted for the object. The backprojection mask visualizes the support from the wheels of the car. Note that the occluding pedestrian is not part of the backprojection mask. The image has been taken from [32].

### 3.5 Bounding Box Estimation

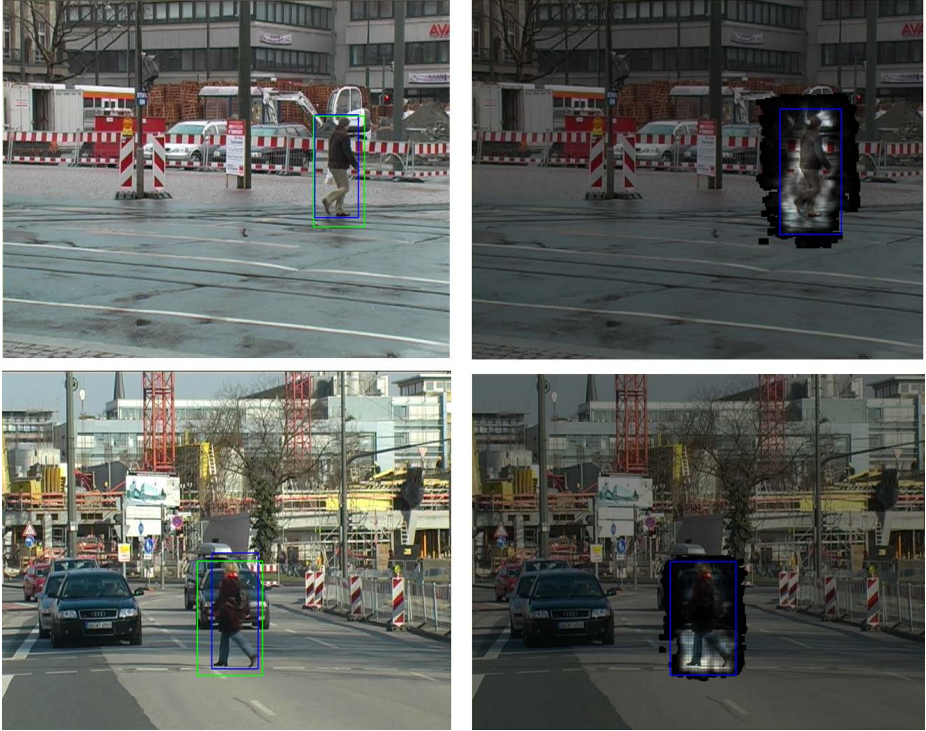
For getting object hypotheses, the modes of  $p(\mathbf{h}|\mathcal{I})$  (11) can be searched by mean shift [23,19] or by smoothing the voting space and searching for local maxima in a greedy manner [16]. In both cases, the bandwidth of the used kernel needs to be large enough to detect objects where the votes do not aggregate in the exact spot. This is illustrated in Figure 7.

The computation of  $p(\mathbf{h}|\mathcal{I})$  can be drastically reduced by sampling not all patches from the test image, but using only a subset a patches. As long as the average overlap between two nearest sampled patches is greater than 50%, the loss in detection performance is acceptable in comparison to the gain in runtime performance [16,18]. In addition, one can discard leaves that are very uncertain as in [8,13,14,19], i.e., if  $p(c|L)$  is low or if the variance of  $p(\mathbf{d}|c, L)$  is high. This can be achieved by using a predefined threshold or taking a fix number of the most certain leaves per image.

Having a hypothesis  $\mathbf{h}(c, \mathbf{x}, s)$ , the enclosing bounding box can be estimated by taking the average bounding box of the training examples of class  $c$ , after rescaling to the unit size  $s_u$ , and multiplying it by the estimated size  $\frac{s}{s_u}$ . The position of the bounding box is defined by  $\mathbf{x}$ .

In some cases, the aspect ratios vary widely within a single class such that the average bounding box of the training images scaled and translated to the detection center is not precise enough. Alternatively, one can compute the backprojection of the supporting image patches for a hypothesis  $\mathbf{h}(c, \mathbf{x}, s)$  [23,32]. One approach to compute the backprojection mask extracts the maximum extent of a possible support, i.e., the largest bounding box of the training images scaled and translated to the detection center. Within the bounding box, the image patches are collected and passed again through the trees. Every time a patch votes for the hypothesis, the contribution weight of the patch  $\mathcal{P}(\mathbf{y})$  for  $\mathbf{h}$  is given by



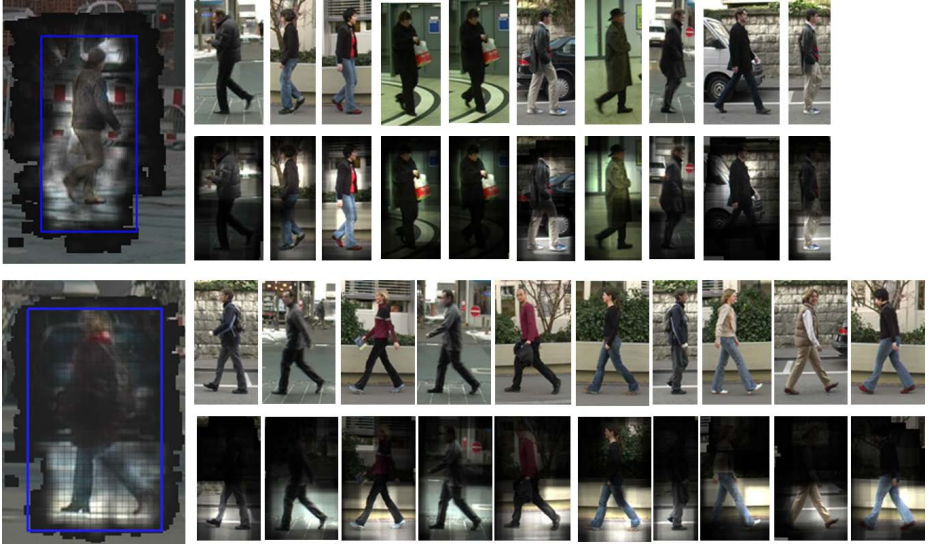


**Fig. 8.** Computing the bounding box based on the backprojection. Left: annotated bounding box (blue) and estimated bounding box (green). Right: Superimposed backprojection mask and estimated bounding box.

$$\pi(\mathbf{h}(c, \mathbf{x}, s), \mathbf{y}) = \frac{1}{|\{T_t\}|} \sum_t \left( \sum_{d \in D_c^{L_t(\mathbf{y})}} \frac{p(c|L_t(\mathbf{y}))}{|D_c^{L_t(\mathbf{y})}|} K \left( d - \frac{s_u(\mathbf{y} - \mathbf{x})}{s} \right) \right), \quad (24)$$

where  $K$  is the kernel used for mode detection. An obtained backprojection mask  $\pi(\mathbf{h}, \mathbf{y})$  is shown in Figure 7. To obtain the bounding box, the mask can be thresholded to estimate the tightest bounding box encompassing the binary mask. In [32], the threshold is defined by  $\frac{1}{2} \max_{\mathbf{y}} \pi(\mathbf{h}, \mathbf{y})$ . Two examples are shown in Figure 8. In [23,34], the backprojection has been additionally augmented by segmentation masks obtained from segmented training data. The segmentation mask can also be used for verification.

In order to detect multiple instances in a single image, one can use a greedy approach. Starting with the hypothesis with the highest score  $p(\mathbf{h}|\mathcal{I})$ , the image patches that support the hypothesis are removed and the detection process continues until a maximum number of hypotheses have been extracted from the image or the remaining hypotheses have a score below a given threshold. However, there are more principled ways to detect multiple instances. In [23,2,34],



**Fig. 9.** Two object hypotheses and their top ten nearest training examples (ordered from left to right). The detected pedestrians are the same as in Figure 8. For each hypothesis, the top row shows the training examples that contribute most to the hypothesis. The bottom row shows the backprojection mask superimposed on each training example. The images have been taken from [32].

optimization procedures for non-maximum suppression, for instance, based on the minimum description length (MDL) principle are used. These methods handle instances that occlude each other better since they aim at solving an optimal assignment of the votes to competing hypotheses.

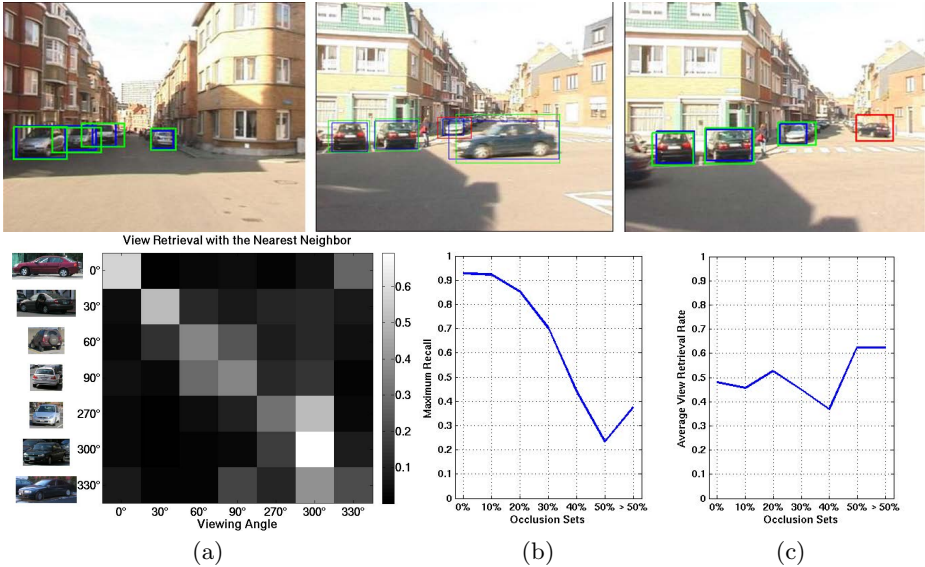
The backprojection can also be used to obtain a link between a hypothesis and the training data [32]. For instance, Equation (24) can be modified by taking only offsets  $D_c^{L_t}(\mathbf{y})(\theta)$  into account that were sampled from a specific training example  $\theta$ . The contribution of a training example for a hypothesis is then measured by  $\sum_{\mathbf{y}} \pi(\mathbf{h}, \mathbf{y}, \theta)$ . Figure 9 shows the training examples that contribute most to the detections shown in Figure 8.

More general, the similarity between two hypotheses  $\mathbf{h}_1$  and  $\mathbf{h}_2$  of the same class  $c$  can be defined by

$$S(\mathbf{h}_1, \mathbf{h}_2) = \frac{\sum_t \sum_{L_t} \sum_{\mathbf{d} \in D_c^{L_t}} \frac{p(c|L_t)}{|D_c^{L_t}|} \mathbb{I}(\mathbf{d}, \mathbf{h}_1) \mathbb{I}(\mathbf{d}, \mathbf{h}_2)}{\sum_t \sum_{L_t} \sum_{\mathbf{d} \in D_c^{L_t}} \frac{p(c|L_t)}{|D_c^{L_t}|} \mathbb{I}(\mathbf{d}, \mathbf{h}_1)} \quad (25)$$

where

$$\mathbb{I}(\mathbf{d}, \mathbf{h}) = \begin{cases} 0 & \text{if } \max_{\mathbf{y}} \pi(\mathbf{h}, \mathbf{y}, \mathbf{d}) = 0, \\ 1 & \text{otherwise.} \end{cases} \quad (26)$$



**Fig. 10.** Viewpoint retrieval on the Leuven car dataset [22]; some examples are shown in the top row (blue - ground truth, green - correct detection, red - incorrect detection). (a) Confusion matrix. Most of the confusions appear between neighboring viewpoints. (b-c) The viewpoint retrieval performance with respect to the amount of occlusion. Although the detection performance deteriorates with an increasing amount of occlusion (b), the viewpoint retrieval performance is affected very little (c), which shows the robustness of the similarity measure to occlusions. The images have been taken from [32].

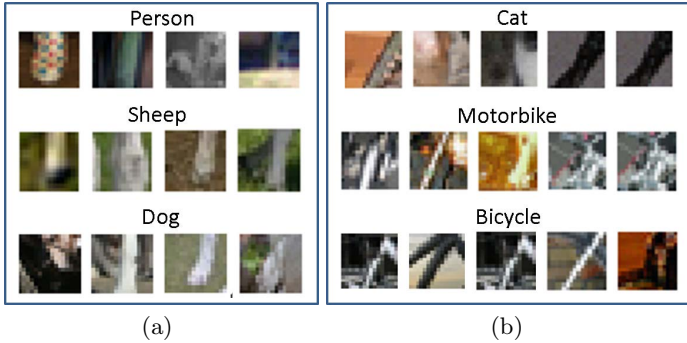
The indicator function  $\mathbb{I}(\mathbf{d}, \mathbf{h})$  is basically 1 if an offset  $\mathbf{d}$  contributes to a hypothesis  $\mathbf{h}$ , which is measured by  $\pi(\mathbf{h}, \mathbf{y}, \mathbf{d})$ , i.e., Equation (24) computed for a single offset  $\mathbf{d}$  instead of  $\sum_{\mathbf{d} \in D_c^{L_t(\mathbf{y})}}$ .

Having a similarity measure, one can retrieve the nearest neighbors from the training set and transfer attributes from them to the detection hypothesis. For instance, the viewpoint of a detected car is estimated using Equation (25) in [32]. The most interesting property of the similarity measure based on the support of two hypotheses is the robustness to occlusions as shown in Figure 10.

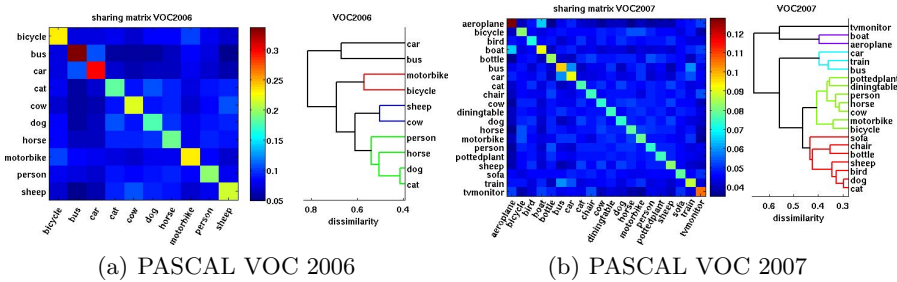
### 3.6 Feature Sharing

The advantage of using one multi-class detector compared to having a detector for each positive class is the ability of sharing features among classes, which reduces the memory requirements and also the testing time. The sharing and the performance of a random forest for multi-class object detection on the PASCAL VOC 2006 and 2007 datasets [12] have been investigated in [33].

The sharing among classes is illustrated in Figure 11. Since each leaf contains patches from several classes, one can compute the amount of sharing among classes [33] by



**Fig. 11.** Patches clustered in two leaves of a multi-class detector trained on PASCAL VOC 2006. The first leaf shares features of similar appearance among the classes person, sheep, and dog. The second example shares features among the classes cat, motorbike, and bicycle. The images have been taken from [33].

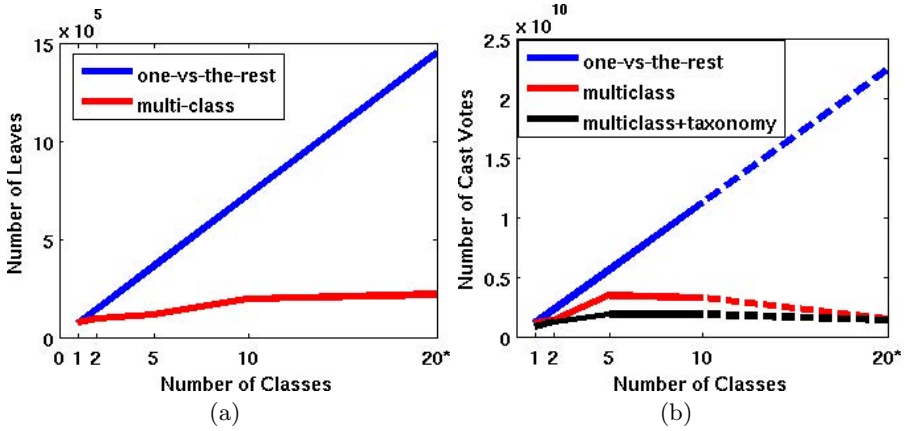


**Fig. 12.** Sharing matrices and their corresponding taxonomies which are automatically obtained by clustering the sharing matrices. The images have been taken from [33].

$$T(c_i, c_j) \propto \sum_t \sum_{L_t} (|D_{c_i}^{L_t}| \cdot p(c_j | L_t)), \quad (27)$$

where  $T(c_i, c_j)$  is normalized such that  $\sum_j T(c_i, c_j) = 1$ . The obtained sharing matrix  $T$  among the positive classes for the two datasets PASCAL VOC 2006 and 2007 are shown in Figure 12. For PASCAL VOC 2006, many features are shared between the pairs bus-car, cat-dog, motorbike-bicycle, and cow-sheep since these categories are also similar in appearance and shape. For dissimilar categories like bus and cow, the sharing is marginal.

Based on the sharing matrix  $T$ , one can derive a taxonomy of classes by clustering the symmetric dissimilarity matrix  $D = 1 - \frac{1}{2}(T + T^T)$ . The automatically derived taxonomies are also plotted in Figure 12. The taxonomies show that the feature sharing within a multi-class random forest is meaningful. The close similarity between cow and sheep can be explained by the typical green background of the training images of the two classes. Since the forest is trained on bounding boxes, many patches with class labels for cow and sheep contain mainly grass



**Fig. 13.** Both the number of leaves (a) and the number of votes (b) of a multi-class random forest grow sublinearly with respect to the number of classes. In contrast, one-vs-the-rest approaches grow linearly. The difference between the blue and the red curve in (a) indicates the amount of sharing that is happening. In (b), the derived taxonomy can be used to further reduce the number of votes. The images have been taken from [33].

from the background. The sharing, however, depends on the image features that are used for the split functions. For instance, potted plant and dining table are measured as similar for PASCAL VOC 2007. Since potted plants are not well described by the used histogram of gradients features, the location of the category in the taxonomy is not very meaningful.

Figure 13 shows the effect of sharing of a multi-class random forest in comparison to training a random forest for each positive class (one-vs-the-rest). Not only the number of leaves is reduced, yielding less memory requirements, but also the votes to be cast for detection is lower. This is achieved by casting only votes if  $p(c_j|L_t) > \frac{1}{C}$ , where  $C$  is the number of classes. Based on the taxonomy, one can even adjust the thresholds for the classes to reduce the number of cast votes further [33].

The multi-class forest can also be used to generate class hypotheses that are verified with a more sophisticated classifier or detector. In [33], the verification detector [15] has been used for re-scoring each hypothesis. The performance on PASCAL VOC 2006 is shown in Table II. The multi-class random forest (MC) and the taxonomy (T) perform similar or better than many one-vs-the-rest (OvA) random forests even after the verification step. While the number of verifications scales well with the number of classes as shown in Table 2, there is no loss in detection performance compared to [15]. As reported in [33], the system requires 35 seconds per image for detecting one positive class, but only 100 seconds for detecting all 20 classes. Comparing these numbers with the fast verification detector [15], which requires 7 seconds per image and per class and 134 seconds per image for 20 classes, there is already a benefit for less

**Table 1.** Performance comparison of a multi-class method (MC) with some baselines in average-precision for the PASCAL VOC 2006 dataset. The first block shows the detection without verification and without non-maxima suppression. MC outperforms one-vs-the-rest (OvA). The taxonomy not only reduces the amount of voting (Figure 13), it also gives a slight improvement. In the second block, verification is performed with 15. By using a two-stage method, there is no loss in accuracy compared to 15. The number of performed verifications is given in Table 2.

Method	bic.	bus	car	cat	cow	dog	hrs.	m.bi.	pers.	shp.	avg
OvA	.16	.13	.07	.04	.18	.03	.15	.16	.11	.12	.114
MC	.37	.12	.11	.02	.14	.05	.08	.21	.05	.12	.127
MC+T.	.38	.13	.12	.05	.15	.03	.11	.12	.05	.12	.132
15	.64	.62	.634	.23	.46	.14	.45	.61	.38	.45	.459
OvA+vrf.	.67	.62	.62	.23	.46	.14	.46	.62	.35	.43	.461
MC+vrf.	.68	.64	.65	.20	.47	.14	.44	.64	.38	.43	.465
MC+T.+vrf.	.66	.64	.66	.22	.47	.14	.44	.64	.36	.42	.463

**Table 2.** The multi-class random forest (MC) reduces the number of windows for verification per image. Since the hypotheses already have a class label, each hypothesis or window needs to be verified only once. It is important that the reduction is achieved without compromising accuracy; see Table 1.

Method	#windows	#verifications
MC-VOC'06 (10 cat.)	1321	1321
MC-VOC'07 (20 cat.)	1778	1778
15-VOC'07 (20 cat.)	42278	833141

than 20 classes. Although it is clear that 100 seconds are still not satisfying, optimizing the random forest for multi-class object detection as in 19 or using the approximations mentioned in this paper might give a significant reduction of the detection time.

## 4 Discussion and Conclusion

In this paper, we have described a general random forest framework for multi-class object detection and discussed several implementation variations. In this context, object detection is formulated as a combined regression and classification problem. While the detection problem becomes a distribution estimation problem, the random forests allow to learn features and descriptors that are optimal for estimating the distributions with low uncertainty. The theoretical framework, however, has the shortcoming that general distributions become too expensive for large datasets. Therefore, several approximations have been discussed to improve the efficiency. The approximations range from restricting the type of distributions to Gaussians or Gaussian mixture models to using an approximation of the spatial distribution for measuring the gain or using sub-sampling strategies during training and testing. Although many approximations

are very intuitive and the basic algorithm is straightforward to implement, it requires some engineering to find an optimal trade-off between accuracy and runtime performance. The most crucial parameter for the detection accuracy, however, is the amount of training data. Random forests are not designed to generalize from small training sets, but to handle large amount of training data efficiently. For datasets with limited training data and large intra-class variation like PASCAL VOC 2007, they do not achieve the best detection accuracy without an additional verification step [18,33]. However, using semi-supervised learning and regularizers that exploit large amount of unlabeled data as described in this paper might overcome the overfitting problem of random forests partially. Due to its relation to implicit shape models [23], the detection approach shares advantages and limitations of this type of models. While techniques like back-projection and feature sharing allow to reason about object hypotheses and the similarity of categories, which goes beyond black box classifiers, the independent assumption of the image patches is a weakness of these models that needs to be addressed in the future. Nevertheless, random forests have a strong potential for applications where many labeled examples are available. For instance, pose or body part estimation from depth data [13,19] are examples where accurate results can be obtained in real-time. The work [19] also shows the benefit of engineering where a fine tuned version of [18] resulted in a speed-up by a factor of 3200.

## References

1. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. *Neural Computation* 9(7), 1545–1588 (1997)
2. Barinova, O., Lempitsky, V., Kohli, P.: On the detection of multiple object instances using hough transforms. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2010)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Computer Vision and Image Understanding* 110(3), 346–359 (2008)
4. Bosch, A., Zisserman, A., Muñoz, X.: Image classification using random forests and ferns. In: *Int'l Conf. Computer Vision* (2007)
5. Breiman, L.: Random forests. *Machine Learning* 45(1), 5–32 (2001)
6. Chen, H.-T., Liu, T.-L., Fuh, C.-S.: Segmenting Highly Articulated Video Objects with Weak-Prior Random Forests. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3954, pp. 373–385. Springer, Heidelberg (2006)
7. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Tech. Rep. MSR-TR-2011-114*, Microsoft Research, Cambridge (2011)
8. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in ct studies. In: *Medical Computer Vision Workshop* (2010)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 886–893 (2005)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2009)

11. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Analysis and Machine Intelligence* (2012)
12. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2), 303–338 (2010)
13. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2011)
14. Fanelli, G., Weise, T., Gall, J., Van Gool, L.: Real time head pose estimation from consumer depth cameras. In: *Pattern Recognition*, pp. 101–110 (2011)
15. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Analysis and Machine Intelligence* 32, 1627–1645 (2010)
16. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2009)
17. Gall, J., Razavi, N., Van Gool, L.: On-line adaption of class-specific codebooks for instance tracking. In: *British Machine Vision Conf.* (2010)
18. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 33, 2188–2202 (2011)
19. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient regression of general-activity human poses from depth images. In: *Int'l Conf. Computer Vision* (2011)
20. Godec, M., Roth, P., Bischof, H.: Hough-based tracking of non-rigid objects. In: *Int'l Conf. Computer Vision* (2011)
21. Lehmann, A., Leibe, B., Van Gool, L.: Fast prism: Branch and bound hough transform for object class detection. *Int'l J. Computer Vision* 94, 175–197 (2011)
22. Leibe, B., Cornelis, N., Cornelis, K., Van Gool, L.: Dynamic 3d scene analysis from a moving vehicle. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2007)
23. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. *Int'l J. Computer Vision* 77(1-3), 259–289 (2008)
24. Leistner, C., Godec, M., Schulter, S., Saffari, A., Werlberger, M., Bischof, H.: Improving classifiers with unlabeled weakly-related videos. In: *IEEE Conf. Computer Vision and Pattern Recognition* (2011)
25. Leistner, C., Saffari, A., Santner, J., Bischof, H.: Semi-supervised random forests. In: *Int'l Conf. Computer Vision*, pp. 506–513 (2009)
26. Lepetit, V., Laguerre, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 775–781 (2005)
27. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91 (2004)
28. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 34–40 (2005)
29. Menze, B.H., Kelm, B.M., Splitthoff, D.N., Koethe, U., Hamprecht, F.A.: On Oblique Random Forests. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) *ECML PKDD 2011*. LNCS, vol. 6912, pp. 453–469. Springer, Heidelberg (2011)
30. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: *Neural Information Processing Systems* (2006)



31. Okada, R.: Discriminative generalized hough transform for object detection. In: Int'l Conf. Computer Vision (2009)
32. Razavi, N., Gall, J., Van Gool, L.: Backprojection Revisited: Scalable Multi-view Object Detection and Similarity Metrics for Detections. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 620–633. Springer, Heidelberg (2010)
33. Razavi, N., Gall, J., Van Gool, L.: Scalable multi-class object detection. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 1505–1512 (2011)
34. Rematas, K., Leibe, B.: Efficient object detection and segmentation with a cascaded hough forest. In: IEEE Workshop on Challenges and Opportunities in Robot Perception (2011)
35. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. In: British Machine Vision Conf. (2008)
36. Schulter, S., Leistner, C., Roth, P., Bischof, H., Van Gool, L.: On-line hough forests. In: British Machine Vision Conf. (2011)
37. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: IEEE Conf. Computer Vision and Pattern Recognition (2011)
38. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: IEEE Conf. Computer Vision and Pattern Recognition (2008)
39. Sun, M., Bradski, G., Xu, B.-X., Savarese, S.: Depth-Encoded Hough Voting for Joint Object Detection and Shape Recovery. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 658–671. Springer, Heidelberg (2010)
40. Winn, J., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 37–44 (2006)

# Segmentation and Classification of Objects with Implicit Scene Context

Jan D. Wegner<sup>1</sup>, Bodo Rosenhahn<sup>2</sup>, and Uwe Sörgel<sup>1</sup>

<sup>1</sup> Institute of Photogrammetry and GeoInformation

<sup>2</sup> Institut für Informationsverarbeitung  
Leibniz Universität Hannover

**Abstract.** We present a novel approach to segment and classify objects in images into two classes. A binary conditional random field (CRF) framework is augmented with an unsupervised clustering step learning contextual relations of objects, the so-called implicit scene context (ISC). Several experiments with simulated data, images from benchmark data sets, and aerial images of an urban area show improved results compared to a standard CRF.

**Keywords:** segmentation, classification, conditional random field, context, clustering.

## 1 Introduction

Object segmentation and classification in images is of major interest in such different fields of science as computer vision, medical vision, and remote sensing. The general aim is to assign a class label to each pixel of an image as opposed to object detection, where rectangular bounding boxes are drawn around an object. In case evidence directly at an object is insufficient to decide on an appropriate class label, contextual information of a characteristic neighbourhood can support segmentation and classification [18]. For example, a building facade often appears in an image with some sky above and street below. Knowing this typical ordering of objects can help distinguishing them.

One way to learn and infer contextual structures in images is to use graphs, which represent different parts of an image as so-called nodes being linked via edges. Characteristic contextual relations between image parts can be captured through edges thus supporting classification. A probabilistic way to exploit graphs for classification combining direct object evidence and context is random fields.

This paper gives a detailed explanation of an approach to contextual binary object classification based on Conditional Random Fields (CRF) originally presented in [28]. It includes all its contribution, but adds experiments with aerial images containing highly complex urban scenes. In the following, a comprehensive review of related work is provided before turning to a brief introduction to standard pair-wise CRFs. A graph structure based on image super-pixels is described and a new way of context modelling is introduced called implicit scene

context (ISC). Its performance is critically evaluated on images of different context complexity levels. Finally, conclusions are drawn and ideas for contextual inference of highly complex scenes are presented.

## 1.1 Related Work

Lafferty and collaborators proposed Conditional Random Fields [4] to label sequential data. CRFs are contextual graphical models like Markov Random Fields (MRF), but provide higher modelling flexibility for classification tasks. Kumar and Hebert extended CRFs to two-dimensional data and applied them to object detection in images [6]. They consider contextual knowledge through pair-wise potentials weighted with features.

He et al. [9] learn pairwise relationships between parts of an image at multiple scales. Local, regional and global features are generated and combined within a single CRF. They may thus capture topologies of scenes at various scales from fine details at a very local level to coarse scene structures of the entire image. In [7] Kumar and Hebert propose a similar approach designing a CRF with two layers. The first layer learns pair-wise relationships between different classes at pixel-level, the second layer captures dependencies between super-pixels. Super-pixels are rather large and typically the image is partitioned into approximately twenty super-pixels. This way the CRF can learn both the global distribution of object classes within a scene and local relationships of object class details. Such approach works well on small images with clearly observable scene structures consisting of few classes of large objects.

In general, CRFs provide a highly flexible framework for contextual classification approaches. Torralba et al. [12] use Boosting to learn contextual knowledge within a CRF framework. Spatial arrangements of objects in an image are learned by a weak classifier and object detection and image segmentation are done simultaneously. Shotton et al. [14] propose a similar concept (but relying on features derived from texton maps) they call "TextonBoost" to achieve joint segmentation and object detection applying boosting within a CRF framework. Lempitsky et al. [32] interleave multi-scale segmentation and object recognition probabilistically without considering any high-level contextual potentials, but using exact and efficient inference.

Another way of directly incorporating contextual knowledge into random fields is to learn whether particular objects or object parts often co-occur in the same scenes and if they have some typical relation. Characteristic spatial distributions of object classes can directly be captured via co-occurrence matrices as, for example, proposed by Carbonetto et al. [25]. The authors learn co-occurrences of objects within a Markov Random Field framework. They test their approach on both a regular grid of square image patches and on super-pixels. Rabinovich et al. [17] propose a similar approach, but formulate a CRF instead of a Markov Random Field. They encode co-occurrence preferences of objects over pair-wise object categories based on image super-pixels. It allows them to distinguish between object categories that often appear together in the same image and, more important, categories that do usually not occur within the same scene.

Galleguillos et al. [23] develop this method further by introducing contextual interactions at pixel-level and at region-level in addition to semantic object interactions via object class co-occurrences. Gould et al. [24] do not solely rely on occurrences, but add a spatial component by modelling relative locations between two object classes and introducing them into a CRF as a unary potential.

In general, all previously reviewed approaches compare pairs of nodes in the CRF graph structure. Functions relating nodes do not deal with more than two nodes at a time. Kohli et al. [22] generalize this classical pair-wise model to higher order potentials that enforce label consistency inside image super-pixels. It allows to model interactions between multiple nodes, functions relate groups of nodes instead of only two. They combine multiple segmentations generated with an unsupervised segmentation method within a CRF for object extraction. Related works of Ladicky et al. [26] propose a hierarchical CRF integrating features computed in different spatial units as pixels, image super-pixels, and groups of super-pixels. They formulate unary potentials over pixels and super-pixels, pair-wise potentials between pixels and between super-pixels and also a connective potential between pixels and the super-pixels they are contained in. All these hierarchical graph-based approaches call for very sophisticated, computationally expensive optimization procedures. Munoz et al. [31] also propose a hierarchical approach, but bypass a global probabilistic model by training separate classifiers at different levels of a multi-scale segmentation.

## 1.2 Contribution

The implicit scene context-CRF (ISC-CRF) for binary object classification originally proposed in [28] is explained in detail. This paper includes all of the contribution of [28] and adds additional experiments with remote sensing data.

In contrast to all reviewed work neither an additional potential is added nor any complex graph structure is generated, but the flexibility provided by the definition of the CRF association potential is exploited thus keeping the global probabilistic model. Context is represented via histograms as done by Belongie et al. [5] and Savarese et al. [13]. Characteristic patterns within the background class of partially labeled images and their relation to labeled object classes are learned. Rotation invariance is achieved and the use of multiple context scales ensures good performance for both small and big objects.

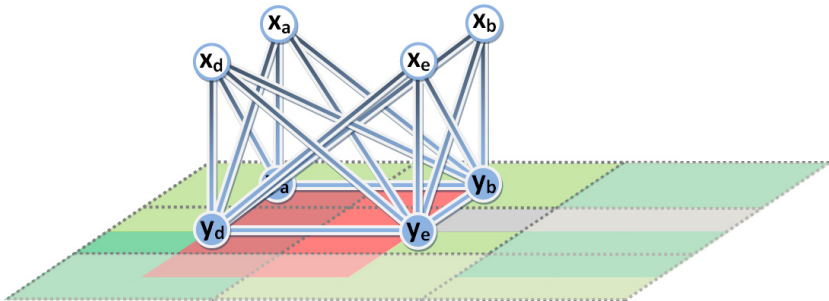
Although the implicit scene context is modelled within a binary CRF framework, it can generally be utilized (with minor changes) with any kind of non-contextual classifier like Support Vector Machines, too. Furthermore, it is generally applicable to arbitrary image scenes, for example, aerial, terrestrial, and medical images.

## 2 Research Background: Conditional Random Fields

In this section we give a short introduction to the basic theory of standard Conditional Random Fields with a pair-wise factorization (Fig. 1). CRFs belong

to the family of undirected graphical models being closely related to Markov Random Fields (MRF). CRFs were originally introduced by Lafferty et al. [4] to label one-dimensional text sequences. Kumar and Hebert [6,15] extended CRFs to two-dimensional data to label images<sup>1</sup>.

CRFs are discriminative techniques meaning that they directly model the posterior distribution  $P(\mathbf{y}|\mathbf{x})$  of the labels  $\mathbf{y}$  given data  $\mathbf{x}$  as a Gibbs distribution as opposed to MRFs being generative methods modelling the joint probability  $P(\mathbf{x}, \mathbf{y})$ . Thus, a CRF can also be viewed as an incomplete model  $P(\mathbf{y}|\mathbf{x})$  whereas an MRF is a complete model  $P(\mathbf{x}, \mathbf{y})$ . CRFs are globally conditioned on all data and we can thus design potential functions relating data of arbitrary locations in an image. In figure 1 a CRF of an example graph is shown. For instance, label  $y_d$  of node  $d$  is not only connected to its own data  $x_d$ , but also to the data of all other nodes  $x_a, x_b, x_d$ , and  $x_e$ . In the prior term node labels are compared with respect to data, too. In the following, it is described how these properties can be expressed more formally.



**Fig. 1.** Labeling of nodes with labels  $\mathbf{y}$  that depend on all data  $\mathbf{x}$  globally with a pairwise factorized CRF (only a subset of the nodes is shown for visualization purposes)

We have an energy term  $E(\mathbf{x}, \mathbf{y})$  encapsulating unary and pair-wise parts. Potential functions of CRFs do not necessarily have to be formulated as probabilities, but they have to be valued positively. Usually, functions out of the exponential family are used to turn energies into potentials. In order to gain a posterior distribution  $P(\mathbf{y}|\mathbf{x})$ , we need to turn potentials into probabilities by normalizing them through the partition function  $Z(\mathbf{x})$ . We may then write the posterior distribution  $P(\mathbf{y}|\mathbf{x})$  as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(E(\mathbf{x}, \mathbf{y})). \quad (1)$$

Following the notations of Kumar and Hebert [15] we can express the energy term  $E(\mathbf{x}, \mathbf{y})$  as the sum of a first term that associates labels with data  $A_i(\mathbf{x}, y_i)$  and a second term that defines how labels interact (incorporating data)  $I_{ij}(\mathbf{x}, y_i, y_j)$ :

<sup>1</sup> Kumar and Hebert [6] call their method Discriminative Random Fields because they use discriminative functions for both the unary and the pair-wise potentials. This particular choice of the potential functions does not change the general CRF framework and thus we will keep the notation Conditional Random Field here.

$$E(\mathbf{x}, \mathbf{y}) = \sum_{i \in \mathbf{S}} A_i(\mathbf{x}, y_i) + \sum_{i \in \mathbf{S}} \sum_{j \in \mathbf{N}_i} I_{ij}(\mathbf{x}, y_i, y_j) \quad (2)$$

Substituting this energy function into equation [1](#) we get the standard CRF expression for two-dimensional data of the posterior  $P(\mathbf{y}|\mathbf{x})$  of labels  $\mathbf{y}$  conditioned on all data  $\mathbf{x}$  [15](#):

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left( \sum_{i \in \mathbf{S}} A_i(\mathbf{x}, y_i) + \sum_{i \in \mathbf{S}} \sum_{j \in \mathbf{N}_i} I_{ij}(\mathbf{x}, y_i, y_j) \right). \quad (3)$$

The left term of equation [3](#) is also called *association potential*  $A_i(\mathbf{x}, y_i)$ . It measures how likely a node  $i$  is labeled with  $y_i$  given all data  $\mathbf{x}$ .  $I_{ij}(\mathbf{x}, y_i, y_j)$  is also referred to as the *interaction potential* and it defines how the labels of two nodes  $i$  and  $j$  interact. As previously explained, both potentials have access to the whole image. In particular the interaction potential  $I_{ij}(\mathbf{x}, y_i, y_j)$  is not only a function of adjacent labels  $y_i$  and  $y_j$ , but of all data  $\mathbf{x}$ , too. Neighbourhood  $\mathbf{N}_i$  of node  $i$  may potentially be the entire image. This is convenient if we want to compare labels based on underlying data. In addition, both the association potential and the interaction potential are defined over all data. Therefore, we can introduce both local and global context knowledge. To obtain a posterior probability  $P(\mathbf{y}|\mathbf{x})$  of labels  $\mathbf{y}$  conditioned on data  $\mathbf{x}$ , the exponential of the sum of association potential and interaction potential is normalized by division through the partition function  $Z(\mathbf{x})$ . It has to be evaluated for each new parameter set during training, but is a constant for a given data set once parameters have been adjusted. Our modelling of both  $A_i(\mathbf{x}, y_i)$  and  $I_{ij}(\mathbf{x}, y_i, y_j)$  of the standard CRF is closely related to the approach proposed by [15](#). Both potentials are discriminatively formulated as linear models:

$$A_i(\mathbf{x}, y_i) = y_i \mathbf{w}^T \mathbf{h}_i(\mathbf{x}), \quad I_{ij}(\mathbf{x}, y_i, y_j) = y_i y_j \mathbf{v}^T \mu_{ij}(\mathbf{x}). \quad (4)$$

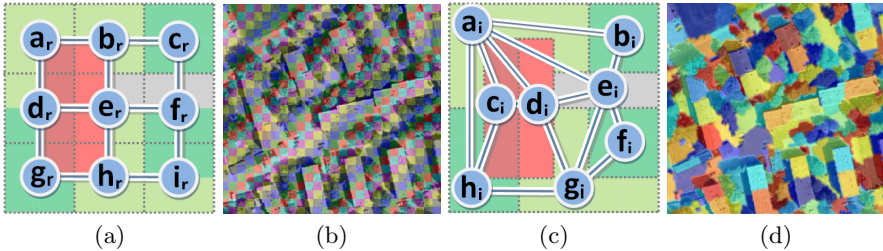
Vector  $\mathbf{h}_i(\mathbf{x})$  contains all node scalar features. Vector  $\mathbf{w}^T$  contains weights of features in  $\mathbf{h}_i(\mathbf{x})$  that are tuned during the training process. Features that help to discriminate the object classes receive high weights, whereas those that do not considerably contribute are down-weighted. In the interaction potential  $I_{ij}(\mathbf{x}, y_i, y_j)$  the comparison of labels  $y_i$  and  $y_j$  follows the Ising model  $\beta y_i y_j$  because we deal with a binary classification task. With  $\beta = 1$ , the product  $y_i y_j$  becomes -1 if labels  $y_i$  and  $y_j$  do not belong to the same class, whereas their product is 1 in case both labels are equal.

### 3 Conditional Random Fields on Super-Pixels

The ISC-CRF, which will be explained in detail in the following section, builds upon an irregular graph of super-pixels of arbitrary shape. He et al. [9](#) were the first to combine graphs of super-pixels with CRFs for object detection. More sophisticated contextual learning based on super-pixels was published by, for

example, Kohli et al. [22] and Gould et al. [24]. Pros and cons of this graph layout are reviewed in this section.

Representing each pixel of an image with a node in the graph is infeasible for large images and datasets because training and inference become computationally very expensive. A standard principle to reduce graph size and computational costs is to divide an image into a grid of square image patches (e.g., [15]) (Fig. 2(a)). A patch grid is set up independently of the scene content following the image grid structure. It does not consider objects contained in the image, therefore patches often cut across boundaries of objects (Fig. 2).



**Fig. 2.** (a) Regular graph on image patches in a 4-connectivity neighbourhood, (b) image patch grid overlaid to aerial photo, (c) irregular graph on image super-pixels, (d) image segmentation overlaid to the same optical aerial image as in (b)

A graph based on super-pixels preserves object boundaries, the structure of the scene is expressed via the graph structure, its size is usually significantly reduced thus decreasing computation time [30], and expressive context formulation is facilitated. It should be noted that an additional advantage of super-pixels is that they capture object shapes enabling the introduction of features like shape, size, main orientation, and roundness.

Super-pixel-graphs call for a particular treatment because they have an irregular structure, defined by the segmentation (Fig. 2c,d), where nodes have different numbers of neighbours as opposed to the regular patch grid. In case of a regular grid of image patches all nodes have equal numbers of neighbouring nodes (except those at image boundaries and in corners). Depending on the connectivity of the neighbourhood, either four or eight, nodes have four or eight edges, respectively. If setting up an irregular graph of super-pixels, the number of adjacent nodes and edges differs significantly depending on the image content and the applied segmentation technique.

Considering nodes  $a_i$  and  $b_i$  in figure 2(c), node  $a_i$  has five neighbours whereas  $b_i$  only two. Nodes with many neighbours would gain a higher weight than nodes with less neighbours. In addition, the impact of the association potential of a node on its label will significantly decrease the more neighbours exist. The label of node  $a_i$  would basically become a function of its neighbouring nodes, its own features would significantly lose importance. A very high number of edges would lead to the label of that node being almost independent of its association potential. In order to avoid this bias some regularization has to be introduced.

For example, Fulkerson et al. [27] use the shared boundary length between two super-pixels as a regularizer thus generally giving more weight to adjacent super-pixels that share longer boundaries. We put equal weights on all neighbours because in our experience a longer shared boundary is not always an indicator for higher significance. Therefore, we normalize each edge feature vector  $\mu_{ij}(\mathbf{x})$  through the sum of norms of feature vectors  $\mathbf{h}_j(\mathbf{x})$  of neighbouring nodes at that particular node to obtain  $\mu_{ij,irregular}(\mathbf{x})$ :

$$\mu_{ij,irregular}(\mathbf{x}) = \mu_{ij}(\mathbf{x}) / \sum_{j \in N_i} |\mathbf{h}_j(\mathbf{x})|. \quad (5)$$

In this way it is guaranteed that no priority is given to nodes with more neighbours and all nodes have per se equal weighting. It is noteworthy that this graph of super-pixels is anisotropic in contrast to the patch graph. The value of an edge potential between two nodes in the super-pixel graph depends on its direction, whereas this is not the case for the isotropic regular graph of image patches. In figure 2(c), the edge between  $h_i$  and  $a_i$  receives another weighting than vice versa, for example, because  $a_i$  has five neighbours and  $h_i$  only three.

## 4 Implicit Scene Context

In this section a method is described integrating data globally, thus exploiting the definition of  $A_i(\mathbf{x}, y_i)$  to its full extent. Even though computing features in several resolutions enlarges a specific local neighbourhood beyond the capabilities of MRFs, most of the techniques (e.g., [9,15]) rest quite local. The definition of CRFs allows to consider all data  $\mathbf{x}$  in association and interaction potential, no restrictions exist with respect to location or correlation of features.

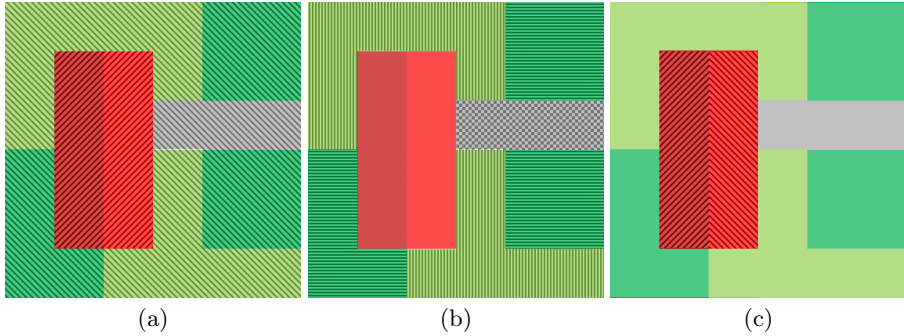
The key idea of this approach is to capture contextual relations of sub-categories of labeled classes in training data, the so-called *implicit scene context* [28]. We call this constellation of sub-categories *implicit* because no explicit semantic object category is assigned during classification.

For example, training data is labeled with the two classes *building* and *background* as shown in figure 3(a). Background consists of several implicit sub-categories like dark green tress, light green grass, and grey driveway (Fig. 3(b)). Class building contains the two implicit sub-categories dark red roof plane and light red roof plane (Fig. 3(c)).

The idea of ISC is to learn characteristic spatial patterns of those implicit sub-categories to support object classification without giving semantics to each object explicitly. This implicit context formulation allows to not explicitly know all object classes contained in the data for training. Moreover, the context level of detail can be chosen by a parameter of the algorithm instead of having to label all training data again if a more detailed scene description is required.

The following requirements have to be met by the algorithm: It should be able to cope with very local to global context scales. In addition, ISC shall be kept generically applicable to multiple kinds of scenes. It should capture, for instance, context in terrestrial images of building facades, where usually sky is above the





**Fig. 3.** Example: (a) two classes building and non-building, (b) implicit object categories contained in background class, (c) implicit categories contained in building class

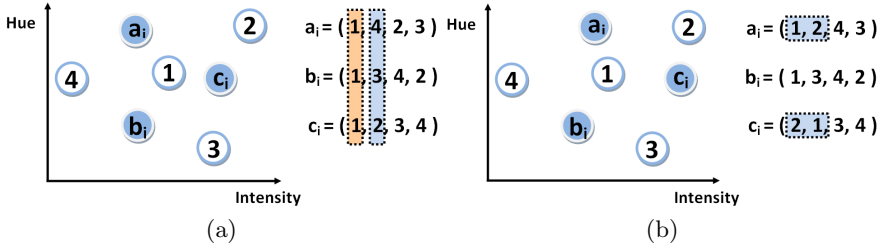
facade and vegetation below, but also in aerial images of buildings, where no preferred ordering with attributes like "above" and "below" exists. Thus, no preferred direction should be relied on. Finally, computational efficiency shall be achieved and computation of co-occurrences be avoided. All steps necessary for training will be explained next followed by a description of the testing phase. In order to meet the requirements aforementioned, training consists of:

- Multi-scale segmentation of images into super-pixels,
- computation of features per super-pixel in all scales,
- unsupervised k-means clustering based on the previously generated features,
- generation of implicit context histograms in three different ranges per super-pixel,
- computation of histogram features,
- integration as feature vector into the CRF unary potentials,
- and training of the CRF based on labeled images.

An unsupervised classification of all super-pixels is performed first for training. Any kind of unsupervised classifier could be applied, but for means of speed and simplicity a standard k-means clustering is chosen. As input to k-means clustering all features  $\mathbf{h}_i(\mathbf{x}) \in \mathbf{h}(\mathbf{x})$  computed per super-pixel are taken. The cluster centers  $\mathbf{K}$  generated with k-means clustering  $\mathbf{K} = K_{means}(\mathbf{h}(\mathbf{x}))$  are used for the following processing.

After k-means clustering, distances to all cluster centers  $\mathbf{K}$  are determined in feature space for each super-pixel. Cluster indices  $\mathbf{y}_{us}$  are recorded in ascending order in a vector per super-pixel according to their distances, the closest center first, the furthest last. Recording not only the closest center, which would correspond to a Minimum Distance classifier, but all others in ascending order, too, has advantages in terms of descriptive context learning and robustness.

Figure 4(a) shows an example consisting of three nodes  $a_i$ ,  $b_i$ , and  $c_i$  in blue circles with white frames in feature space defined by hue and intensity. Cluster centers 1 to 4 computed with k-means (considering additional nodes to the ones shown in Fig. 4) are depicted in white circles with blue frames. Indices 1 to 4 are



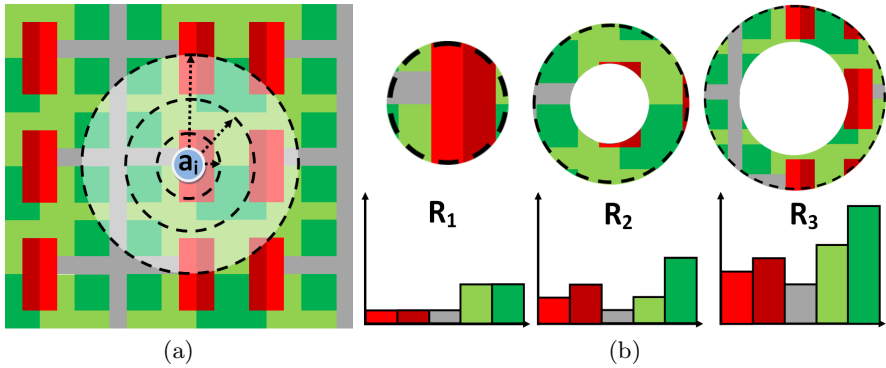
**Fig. 4.** Two-dimensional feature space spanned by hue and intensity: Nodes  $a_i$ ,  $b_i$ , and  $c_i$  and cluster centers 1, 2, 3, and 4 are shown; cluster centers are recorded in descending order with respect to their distances to the nodes; (a) nodes cannot be distinguished based on closest cluster center (first vector elements in orange frame), but on the second closest (second vector elements in blue frame), (b) gain in robustness: although the closest cluster centers of nodes  $a_i$  and  $c_i$  are different, they belong to the same class because any combination of the first two vector elements (framed in blue), no matter their order, is learned to be descriptive

the indices of the cluster centers, the vector of all indices is  $\mathbf{y}_{us}$ . Assuming  $a_i$  and  $c_i$  to belong to object sub-categories and  $b_i$  to a background sub-category, it would be impossible to distinguish them if taking merely the closest cluster index because all three nodes have equal distances to cluster center one. If just recording the closest center (first element in vectors in Fig. 4(a) framed in orange), all nodes would be labeled one, although they occur at different positions in feature space. The second closest cluster center (framed in blue) is different for all nodes and helps distinguishing.

In order to explain the gain in robustness, figure 4(b) shows a slightly different setup. Nodes  $a_i$  and  $c_i$ , sharing the same class, have distinct closest cluster centers. Nonetheless, considering in addition the second closest elements, too, both nodes share the same first two cluster centers (framed in blue), only their order changes. A feature is defined that accounts for this variation of absolute ordering. Superpixels are considered to be located closely in feature space if the first two vector elements are equal, no matter their order. In conclusion, benefits are twofold: First, the type of cluster centers at each node carries valuable information facilitating detailed distinctions between classes, second, robustness is gained if nodes of the same class are assigned to equal cluster centers, but in different orders.

An example simulating an aerial image of an urban scene is shown schematically in figure 5(a). The task consists of assigning the explicit class labels building or background to each pixel. Class building contains the two implicit sub-categories "light red roof" and "dark red roof" whereas the background class contains the implicit sub-categories "light grey street", "light green grass", and "dark green trees". In total, five distinct sub-categories occur captured with  $k = 5$  cluster centers<sup>2</sup>.

<sup>2</sup> The number of cluster centers has to be set manually a priori. Automatic determination of the exact number of sub-categories in feature space, based on the ISODATA method [19], for example, is left for future work.



**Fig. 5.** Implicit scene context: (a) ranges around the centroid of a super-pixel belonging to implicit sub-category "light red roof" (part of building class) represented by node  $a_i$ , (b) histograms of cluster labels of three ranges  $R_1$ ,  $R_2$ , and  $R_3$ ; the ordinate counts the number of super-pixels per cluster label within a range  $R$ , cluster labels are ordered on the abscissa; colours indicate different cluster labels appointed to super-pixels, boundaries of super-pixels run along colour edges

Next, the centroid  $C_S$  of each super-pixel is determined and histograms of labels  $hist_R(\mathbf{y}_{us})$  occurring within different ranges  $R$  around each super-pixel are generated. Numbers of label occurrences  $\mathbf{y}_{us}$  within a range  $R$  are counted in histograms. This is shown for a node  $a_i$  of sub-category "light red roof" in figures 5(a,b). Occurrences of five different labels are counted in three ranges  $R_1$ ,  $R_2$ , and  $R_3$ <sup>3</sup>. This procedure is conducted for all nodes in the graph.

The entire ISC concept is based on the assumption that histograms of particular sub-categories will have characteristic shapes because neighbouring sub-categories will appear with particular frequencies in certain ranges. Combining histograms of all ranges ( $R_1$ ,  $R_2$ , and  $R_2$  in Fig. 5) results in distinct context distributions of all sub-categories. Either short or long ranges can be chosen depending on whether local or global context is to be integrated. In order to meet the requirements of generalizability and transferability to multiple object classes and scenes, the exact ranges should be adapted to the scale of the context. The scale of the desired object class and its context can be approximated via the size of super-pixels after (over-)segmentation. Ranges  $R$  as a linear function of the mean super-pixel size were found to be optimal after tests with different image data and scenes.

Various moments and additional information representing contextual patterns in the environment of a particular super-pixel are derived from the histograms. It is noteworthy that label histograms can either be directly introduced to node feature vectors or specific features can be derived from histograms, the index of the most often appearing label within each range, the index of the label

<sup>3</sup> Any number of ranges can be chosen depending on the scene and on the scale of context. However, more ranges lead to increasing computational costs; three ranges are usually sufficient.

covering the largest area, for example. Qualitative, quantitative, and spatial context features  $\mathbf{C}(\mathbf{h}(\mathbf{x}))$  can be generated.

For the testing phase, exactly the same processing steps are applied except k-means clustering (and CRF training). Those cluster centers  $\mathbf{K}$ , originally generated with k-means during training, are used to determine closest cluster centers in ascending order per super-pixel of test data. Cluster indices are determined for all test data nodes (i.e., super-pixels of the test images after segmentation), measuring distances in feature space to cluster centers generated in the training phase.

The class of each super-pixel  $i$  can be derived merely based on implicit context features  $\mathbf{C}_i(\mathbf{h}(\mathbf{x}))$  or local node features  $\mathbf{h}_i(\mathbf{x})$  can be added to the feature vector, too. The ISC-CRF unary potential is given in equation 6. Pair-wise potentials only change in such a way that the element-wise absolute differences between nodes  $i$  and  $j$  in the graph are computed based on the corresponding implicit context features (Eq. 7).

$$A_i(\mathbf{x}, y_i) = y_i \mathbf{w}^T \mathbf{C}_i(\mathbf{h}(\mathbf{x})) \quad (6)$$

$$I_{ij}(\mathbf{x}, y_i, y_j) = y_i y_j \mathbf{v}^T \mu_{\mathbf{C},ij}(\mathbf{x}), \quad \mu_{\mathbf{C},ij}(\mathbf{x}) = |\mathbf{C}_i(\mathbf{h}(\mathbf{x})) - \mathbf{C}_j(\mathbf{h}(\mathbf{x}))| \quad (7)$$

No normalization of the label count in the histogram is done based on the size of the super-pixels, for example, because tests show that the importance of a super-pixel does not necessarily increase with its size. Small super-pixels can be characteristic context features and are of high relevance for a particular object class, too. Dealing with a multi-scale segmentation, implicit context histograms can be computed at coarser scales, too. It is possible to learn global context of coarse scene structures at a coarse scale while simultaneously capturing local context at the finest scale<sup>4</sup>.

## 5 CRF Training and Inference

The objective of training is to adjust parameters of the classifier function such that classes are discriminated in an optimal way. In this paper, object detection is viewed as a binary classification (e.g., object versus background), the task is to find an optimal decision surface in feature space separating both classes. Parameters to be trained model shape, orientation, and position of this surface. They are the elements of node weight vector  $\mathbf{w}$  and of edge weight vector  $\mathbf{v}$ . In order to ease notation, one can concatenate parameters of  $\mathbf{w}$  and  $\mathbf{v}$  in a single parameter vector  $\theta = (w_1, w_2, \dots, w_n; v_1, v_2, \dots, v_m)$  with number of node features  $n$  and number of edge features  $m$ . Similarly, feature vectors  $\mathbf{h}(\mathbf{x})$  and  $\mu(\mathbf{x})$  are concatenated to one vector  $\Phi$ .

<sup>4</sup> Graphs of image super-pixels generated with a multi-scale segmentation can also be used directly for classification if object shapes are learned via so-called region ancestries as proposed by Lim et al. [8]. The integration of this promising concept into a CRF framework is left for future work.

Adjustment of parameters  $(\mathbf{w}, \mathbf{v})$  is an unconstrained nonlinear optimization problem that has to search a very large space of parameters. Being an entire research area of its own and since focus of this contribution is on context modelling and not on designing optimization techniques, a state-of-the-art method as used in [16] is applied. It couples the optimization method *Limited Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)* [21] with inference via *Loopy Belief Propagation (LBP)* [3] for training. A detailed description of both methods is beyond the scope of this paper, details can be found in the given references.

$P(\mathbf{y}|\Phi, \theta)$  is the posterior probability of the CRF and  $P(\theta)$  is a prior over parameters  $\theta$  acting as a regularization term. It penalizes large parameters thus smoothing the objective function to avoid over-fitting to training data. Usually, the assumption is made that parameters  $\theta$  follow an isotropic Gaussian prior [34] and one may thus write their probability as  $P(\theta) = \exp\left(\frac{-(\theta-\theta_0)^2}{2\sigma^2}\right)$  with  $\theta_0 = 0$ . This leads to a regularization term containing euclidean norm  $\|\theta\|$  of parameters and variance  $\sigma^2$ :

$$P(\theta) = \exp\left(-\frac{1}{2\sigma^2}\|\theta\|^2\right) \quad (8)$$

The choice of  $\sigma$  steers smoothness of the objective function with respect to training data. A larger  $\sigma$  results in a smoother function whereas a smaller  $\sigma$  better adapts the objective function to training data, but at the risk of over-fitting. An appropriate objective function, ensuring exactly one global optimum, has to be designed. It should either be convex (global minimum) or concave (global maximum). A concave objective function can be reformulated as a convex function and vice versa. This criterion is met using the *regularized log likelihood* as objective function [4, 6, 34]. The objective function  $L(\theta)$  to be optimized for parameter estimation is the negative regularized log-likelihood:

$$L(\theta) = -\log(P(\mathbf{y}|\Phi, \theta) \cdot P(\theta)) \quad (9)$$

## 6 Experiments

In order to assess benefits and limitations of the ISC-CRF several experiments are conducted. First, standard CRF (as described in sections 2 and 3) and ISC-CRF (section 4) are compared using a simulated test scene with context of low complexity, where the exact number of sub-categories is known a priori. Second, consequences of varying numbers of k-means cluster centers are investigated. Third, robustness to noise in comparison to the standard CRF is tested. Fourth, experiments with images of building facades taken from the eTrims dataset [33] and with images of algae downloaded from the internet are done. Both kinds of images represent context of medium complexity. Finally, buildings in aerial images of an urban scene showing context of very high complexity are segmented and classified.

Segmentation of images into super-pixels is done with Quickshift [21, 20]. It is particularly convenient for a multi-scale approach because small super-pixels at

fine scales are always completely contained within larger ones at coarse scales without any overlap at boundaries.

## 6.1 Features

In order avoid biasing evaluation by particularly designed sophisticated features we select very simple ones. Mean of red and green channel (normalized by the length of the RGB vector), hue mean and standard deviation, and saturation mean are found to be descriptive colour features. Additional features are generated based on gradient orientation histograms of the intensity image [11] as already used for detection of building facades [15,29]. Second and third central moments of gradient orientation histograms are used as features. All basic features are scaled between zero and one.

A quadratic expansion of feature vectors  $\mathbf{h}_i(\mathbf{x})$  is done as described by Kumar and Hebert [15], who state that this step may be viewed as a *kernel mapping of the original feature vector into a high dimensional space*. It introduces a quadratic decision surface in feature space capable of more precisely discriminating building nodes from background nodes compared to a simple linear one. The basic idea is that a linear classifier applied in a quadratically expanded feature space will yield a quadratic decision surface in original feature space. Simple linear models can be kept allowing for efficient parameter estimation by introducing a higher order feature space.

A quadratic feature vector contains all original elements, their squares, and pairwise products. Kumar and Hebert [15] mention that this is *equivalent to the kernel mapping of the data using a polynomial kernel of degree two*. Each first component of an expanded node feature vector is set to one in order to accommodate a so-called bias parameter, which is the first element of the corresponding weight vector. Its effect can be interpreted as shifting the decision surface in feature space, exact shape modelling is done by all other parameters.

As ISC features we compute the closest and second closest cluster centers to the node of interest (two features), minimum, maximum, median, and standard deviation of occurring cluster indices at each context range (twelve features in case of three context ranges), often and second most often occurring indices at each range (six features in case of three context ranges).

## 6.2 Evaluation Strategy

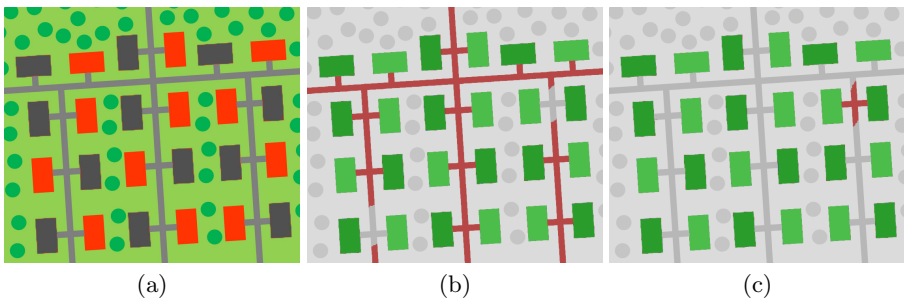
Results of all classification experiments are evaluated in terms of false positive rate (FPR) and true positive rate (TPR). FPR is the percentage of all background pixels being misclassified as building pixels. TPR represents the percentage of all building pixels being correctly classified as such. In general, the goal is to develop a classification technique delivering results with high TPR and low FPR. In order to ease visual interpretability of results, CRF classification outcomes are overlaid to the intensity channel of the optical image. False positive pixels are coloured red, true positive pixels green, missed building pixels blue

(false negatives), and correctly classified background (true negatives) without any colour.

Cross-validation is performed for all experiments in order to avoid particular training/testing-setups biasing classification results. Corresponding to Crowther and Cox [10], the optimum experimental setup is to use two thirds of data for training and one third for testing. Thus, three-fold cross-validation is conducted and each experiment is done three times with changing training and testing image combinations. Twenty ISC features are computed in total at each segmentation scale if considering three context ranges. A multi-scale segmentation with three scales leads to 60 ISC features being written to a node at highest scale.

### 6.3 Simulated Scene of Low Complexity Context

ISC-CRF and standard CRF are first applied to three simulated subscenes (one is shown in Fig. 6(a)) containing red buildings, grey buildings, trees (dark green circles), grassland (light green background), and streets (light grey lines). Only colour features are used because no texture was simulated. Grey buildings and grey streets are closely located in feature space and thus context has to support discriminating buildings from streets. Implicit scene context is captured in three ranges (radii 10, 20, and 30 pixels) and concatenated with original colour features for ISC-CRF classification. Three-fold cross-validation is conducted and mean TPR and FPR are computed. Standard CRF (Fig. 6(b)) and ISC-CRF (Fig. 6(c)) achieve the same TPR of 85.9%. The standard CRF misclassifies 6.8% background pixels as building whereas the ISC-CRF has a significantly lower FPR of 0.8%.

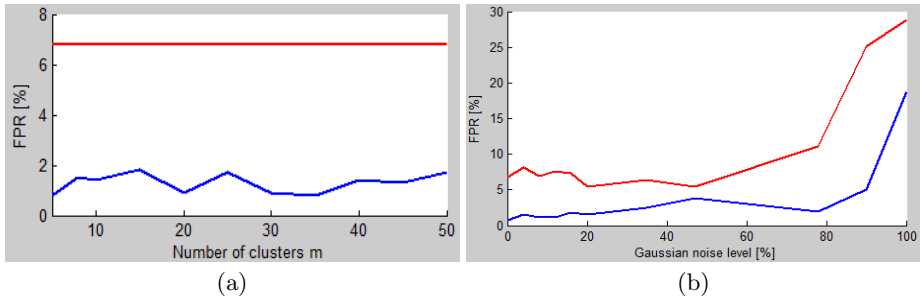


**Fig. 6.** CRF results with simulated data: (a) one image of the simulated test images, (b) detected buildings without implicit scene context (c) and with implicit scene context

As edge feature vector the standard CRF considers the absolute difference of adjacent node feature vectors in order to support or suppress smoothing. Since grey buildings and grey streets are located very closely in colour feature space, the standard CRF cannot well distinguish those two object categories, neither based on node features nor on edge features. It leads to some street super-pixels

being misclassified as building (Fig. 6(b)). The ISC-CRF learns the arrangement of sub-categories "street" and "grey building" (besides all other sub-categories) implicitly and is thus able to discriminate the two. Such being the case, streets are correctly classified as background, although original colour features are not distinctive (Fig. 6(c))<sup>5</sup>. This result shows that scenes with context of low complexity can benefit from implicit scene context.

Cluster center number as well as segmentation scales are currently adapted manually to each data set, whereas context radii are set as a function of the mean super-pixel size of an image. The simulated urban scene (Fig. 6(a)) is used to evaluate the impact of varying cluster centers because the exact number of sub-categories is known: red buildings, grey buildings, trees (dark green circles), grassland (light green background) and streets (light grey lines). Only colour features are used for these tests leading to five distinct clusters. Three ranges (radii 10, 20, and 30 pixels) are chosen and experiments with five up to 50 cluster centers are conducted. FPR of each ISC-CRF classification is displayed in blue Fig. 7(a) and such of the standard CRF in red. The ISC-CRF FPR varies about 1 % (from 0.8 % to 1.8 %) and no significant trend is observable. Changing the number of k-means cluster centers has a very small impact on classification performance, but of course on computation time. A rather small number of cluster centers is beneficial. Segmentation scale is adapted to each scene separately (and context radii are a function of the mean super-pixel size) because it depends on the scales of context and objects. This makes the ISC-CRF highly flexible and easy to adapt to new scenes.



**Fig. 7.** FPR of ISC-CRF (blue) based on simulated data (FPR of standard CRF drawn in red): (a) with varying numbers of cluster centers and (b) with different noise levels (cluster center number fixed to five)

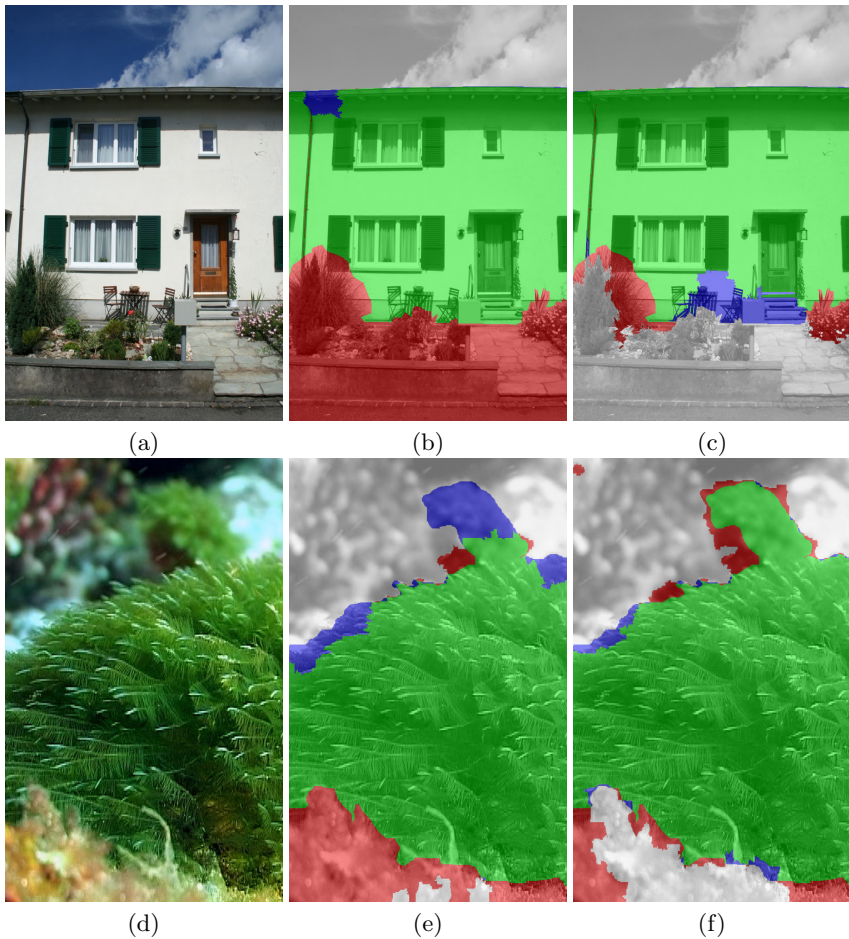
Robustness of the ISC-CRF to noise is experimentally evaluated. Several Gaussian noise levels with mean zero and standard deviations up to 100 % (corresponding to 256 in our case of 8 bit RGB channels) are generated and added to RGB channels of the simulated data, which is then cropped in order to keep

<sup>5</sup> The misclassified part of the street is most probably caused by a boundary effect, which also leads to the same part being correctly classified with the standard CRF in Fig. 6(b).



all values between zero and 255. Cross-validation tests with standard CRF and ISC-CRF are done and FPR is recorded. In figure 7(b) FPR of standard CRF (red) and of ISC-CRF (blue) considering all tested noise levels are displayed. The FPR of the ISC-CRF stays below that of the standard CRF at all noise levels. Furthermore, the ISC-CRF is slightly more robust to noise because its FPR starts increasing later (approx. 90 % vs. approx. 80 %).

Experiments with simulated data show that the general concept of implicit scene context helps discriminating object classes if original features are not distinctive enough. It is robust to noise, even more robust than the standard CRF, and changing the currently manually adjusted number of cluster centers has only a small impact on results.



**Fig. 8.** Comparison of standard CRF (b, e) and ISC-CRF (c, f) for eTrims [33] building facades (a-c) and algae (d-f)

## 6.4 Scenes of Medium Context Complexity

In order to evaluate the applicability of the ISC concept to scenes of medium complexity, tests are performed with two different datasets: Facade images taken from the eTRIMS benchmark data [33] and images of algae downloaded from the internet. Those particular object class categories are chosen because they represent different spatial object and background distributions. Building facades are single very large objects with clear straight boundaries and background context only above and below (Fig. 8(a)). Algae are large but frayed objects partially surrounded by background context (Fig. 8(d)).

Experiments are conducted with nine images of each scene category, which are randomly partitioned into groups of three images for three-fold cross-validation. Example images and corresponding results are shown in figure 8. Classification performance is summarized in table 1.

**Table 1.** TPR and FPR for different objects and context patterns achieved with a standard CRF and with an ISC-CRF

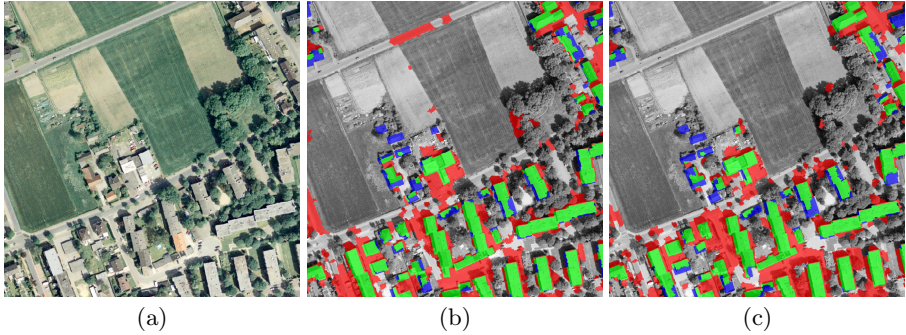
Data	CRF		ISC-CRF	
	TPR [%]	FPR [%]	TPR [%]	FPR [%]
eTRIMS facades	86.9	<b>22.1</b>	88.1	<b>7.3</b>
Algae	75.7	<b>37.0</b>	84.5	<b>23.7</b>

The highest decrease of the FPR (7.3 % vs. 22.1%) is achieved with building facades (Fig. 8(a-c)). Only using the standard CRF, colour and gradient features do not sufficiently well discriminate a building facade from foreground (Fig. 8(b)). However, incorporating implicit scene context (based on the same colour and gradient features), the CRF can well distinguish building facade from vegetation and doorway in the foreground (Fig. 8(c)). A similar result is achieved with the algae images. Implicit scene context decreases the FPR (23.7 % vs. 37.0 %) while increasing the TPR (84.5 % vs. 75.7 %) (cf. Fig. 8(d) & (f)).

## 6.5 Scenes of High Context Complexity

The previous scenes have shown that the concept of the ISC-CRF improves classification results in comparison to such of a standard CRF if applied to images containing scenes with low to medium complexity. In order to investigate the impact of an ISC-CRF on highly complex scenes building classification in aerial images is conducted (Fig. 9). However, results are neither significantly improved nor deteriorated compared to the standard CRF. Compared to a standard CRF, the TPR is slightly higher (79.5% to 76.3%), but the FPR increases (22.4% vs. 20.0%), too.

A reason for this outcome is the highly complex context of the urban scene. Spatial arrangements of sub-categories show a significant variation, which could not sufficiently well be learned by the ISC-CRF. Cluster patterns of buildings and



**Fig. 9.** Classification results for a scene with context of high complexity: (a) Aerial image of an urban area, (b) standard CRF, (c) ISC-CRF (three ranges, five cluster centers)

surrounding sub-categories of the background class are very diverse. This high diversity leads to no significant pattern being learned. One attempt to improve results would be to use much more training data, which calls for setting up a benchmark dataset of high-resolution aerial images. Another way to model relations of sub-categories directly via the graph structure (i.e., pair-wise potentials).

On an Intel<sup>TM</sup> Core i7 2.4 Ghz CPU, 12 GB RAM training and inference per image (of low to medium scene complexity) takes about ten seconds. The implicit scene context potential does only marginally increase computation time by about two seconds per image if dealing with those images (i.e., with a relatively low number of super-pixels below 150 per image). However, implicit scene context significantly slows down training and inference from five minutes per image to ten minutes per image if applied to the highly complex aerial photos (Fig. 9(a)), which are partitioned into approximately 700 super-pixels each.

## 7 Conclusion and Future Work

In conclusion, implicit scene context significantly improves object detection if applied to scenes with context of medium and low complexity. Remote sensing data proves to be the most challenging classification task because context has the highest degree of complexity. Building segmentation and classification is not significantly improved with the ISC-CRF. Therefore, novel ways of sophisticated contextual learning have to be thought of for highly complex scenes.

One possibility could be the integration of a multi-scale segmentation explicitly into the graph resulting in a three-dimensional structure, where messages are passed between super-pixels of neighbouring scales (e.g., [22,23]). Object feature distributions and contextual links could be captured separately at different scene scales. In addition, completely representing a scene topology in multiple scales with a graph would enable inter-scale contextual learning. Region-ancestry concepts as suggested by Lim et al. [8] could be included and re-formulated in a CRF.

Another idea concerning the ISC-CRF is to consider the shapes of super-pixels for context histogram ranges. Instead of simply drawing circular ranges around the super-pixel centroid, one could enlarge the original super-pixel, keeping its shape, by certain ranges. Elongated street super-pixels, for example, sticking out of the first circular range and being counted twice (again in the second range), would be extended by the same distance in any direction thus avoiding double counting. Circular ranges reach out further into the image perpendicularly to an elongated super-pixel, with respect to its boundaries, than lengthwise. Introduction of shape would avoid this bias and give equal importance to any direction.

In general, the CRF prior has not been used to explicitly learn contextual relations of object categories, yet. It basically has stayed a smoothing term. Furthermore, only local to regional context has been learned, yet, although the CRF allows for global context learning. Concerning remote sensing applications one idea would be to use large cartographic databases, for example Open Street Map, to train global contextual relations between urban objects like roads, buildings, and vegetated areas. Learning this global context would be rather fast because cartographic data already exists in vector format. We could exploit very large databases in a relatively short time. Instead of only determining one-by-one relations of the node of interest to a neighboring node we could think of detecting particular context constellations. The association potential of the CRF framework would then learn local object features, the interaction potential regional context, and an additional potential global patterns via cartographic data.

## References

1. Liu, D.C., Nocedal, J.: On the Limited Memory BFGS method for large scale optimization. *Mathematical Programming* 45, 503–528 (1989)
2. Nocedal, J.: Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation* 35, 773–782 (1980)
3. Frey, B.J., MacKay, D.J.C.: A Revolution: Belief Trees: Belief Propagation in Graphs With Cycles. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) *Advances in Neural Information Processing Systems*, pp. 479–485. MIT Press, Cambridge (1998)
4. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for segmenting and labeling sequence data. In: *ICML*, p. 8 (2001)
5. Belongie, S., Malik, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(24), 509–522 (2002)
6. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: *ICCV*, vol. 2, pp. 1150–1157 (2003)
7. Kumar, S., Hebert, M.: A Hierarchical Field Framework for Unified Context-Based Classification. In: *ICCV*, vol. 2, pp. 1284–1291 (2005)
8. Lim, J.J., Arbelaez, P., Gu, C., Malik, J.: Context by Region Ancestry. In: *ICCV*, pp. 1978–1985 (2009)
9. He, X., Zemel, R.S., Carreira-Perpiñán, M.: Multiscale Conditional Random Fields for Image Labeling. In: *CVPR*, p. 8 (2004)

10. Crowther, P.S., Cox, R.J.: A Method for Optimal Division of Data Sets for Use in Neural Networks. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3684, pp. 1–7. Springer, Heidelberg (2005)
11. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, p. 8 (2005)
12. Torralba, A., Murphy, K.P., Freeman, W.T.: Contextual Models for Object Detection Using Boosted
13. Savarese, S., Winn, J., Criminisi, A.: Discriminative Object Class Models of Appearance and Shape by Correlators. In: CVPR, p. 8 (2006)
14. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: *TextonBoost*: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
15. Kumar, S., Hebert, M.: Discriminative Random Fields. *International Journal of Computer Vision* 68(2), 179–201 (2006)
16. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: HLT-NAACL, pp. 213–220 (2003)
17. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in Context. In: ICCV, p. 8 (2007)
18. Oliva, A., Torralba, A.: The role of context in object recognition. *Trends in Cognitive Sciences* 11(12), 520–527 (2007)
19. Ball, G.H., Hall, D.J.: A clustering technique for summarizing multivariate data. *Systems Research and Behavioral Science* 12(2), 153–155 (1967)
20. Vedaldi, A., Fulkerson, B.: VLFeat: An Open and Portable Library of Computer Vision Algorithms (2008), <http://www.vlfeat.org/>
21. Vedaldi, A., Soatto, S.: Quick Shift and Kernel Methods for Mode Seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 705–718. Springer, Heidelberg (2008)
22. Kohli, P., Ladicky, L., Torr, P.H.S.: Robust Higher Order Potentials for Enforcing Label Consistency. In: CVPR, p. 8 (2008)
23. Galleguillos, C., McFee, B., Belongie, S., Lanckriet, G.: Multi-Class Object Localization by Combining Local Contextual Interactions. In: CVPR, p. 8 (2010)
24. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-Class Segmentation with Relative Location Prior. *International Journal of Computer Vision* 80(3), 300–316 (2008)
25. Carbonetto, P., de Freitas, N., Barnard, K.: A Statistical Model for General Contextual Object Recognition. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 350–362. Springer, Heidelberg (2004)
26. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.S.: Associative Hierarchical CRFs for Object Class Image Segmentation. In: ICCV, p. 8 (2009)
27. Fulkerson, B., Vedaldi, A., Soatto, S.: Class Segmentation and Object Localization with Superpixel Neighborhoods. In: ICCV, p. 8 (2009)
28. Wegner, J.D., Rosenhahn, B., Soergel, U.: Implicit Scene Context for Object Segmentation and Classification. In: Mester, R., Felsberg, M. (eds.) DAGM 2011. LNCS, vol. 6835, pp. 31–40. Springer, Heidelberg (2011)
29. Korč, F., Förstner, W.: Interpreting terrestrial images of urban scenes using discriminative random fields. In: ISPRS Symposium Beijing, vol. 37(B3a), pp. 291–296 (2008)
30. Wegner, J.D., Rosenhahn, B., Soergel, U.: Segment-based building detection with Conditional Random Fields. In: Stilla, U., Juergens, C., Maktav, D. (eds.) 6th IEEE/GRSS/ISPRS Joint Urban Remote Sensing Event, pp. 205–208 (2011)

31. Munoz, D., Bagnell, J.A., Hebert, M.: Stacked Hierarchical Labeling. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 57–70. Springer, Heidelberg (2010)
32. Lempitsky, V., Vedaldi, A., Zisserman, A.: A Pylon Model for Semantic Segmentation. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., Weinberger, K.Q. (eds.) NIPS 2011, vol. 24, pp. 1485–1493 (2011)
33. Korč, F., Förstner, W.: eTRIMS Image Database for Interpreting Images of Man-Made Scenes (2009), [http://www.ipb.uni-bonn.de/projects/etrims\\_db/](http://www.ipb.uni-bonn.de/projects/etrims_db/)
34. Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields for Relational Learning. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning. MIT Press, Cambridge (2006)

# Dense 3D Reconstruction from Wide Baseline Image Sets

Helmut Mayer<sup>1</sup>, Jan Bartelsen<sup>1</sup>, Heiko Hirschmüller<sup>2</sup>, and Andreas Kuhn<sup>1,2</sup>

<sup>1</sup> Institute of Applied Computer Science, Bundeswehr University Munich

<sup>2</sup> Institute for Robotics and Mechatronics, German Aerospace Center (DLR)

**Abstract.** This paper describes an approach for Structure from Motion (SfM) for wide baselines image sets and its combination with the dense Semiglobal Matching (SGM) 3D reconstruction approach. Our approach for SfM relies on given information concerning image overlap, but can deal with large baselines and produces highly precise camera parameters and 3D points. At the core of our contribution is robust least squares adjustment with full exploitation of the covariance information from affine point matching to bundle adjustment. Reweighting for robust adjustment is based on covariance information for each individual residual. We use points detected based on Differences of Gaussians including scale and orientation information as well as a variant of the five point algorithm. A strategy similar to the Expectation Maximization (EM) algorithm is employed to extend partial solutions. The key characteristics of the approach is reliability obtained by aiming at a high precision in every step. The capabilities of our approach are demonstrated by presenting results for sets consisting of images from the ground and from small Unmanned Aircraft Systems (UASs).

## 1 Introduction

Structure from Motion (SfM) from sets of images in combination with dense 3D reconstruction forms a good basis for photo realistic visualization. For example, Leberl et al. [14] show that high quality models can be generated from aerial images, in particular for Microsoft Bing Maps. Leberl et al. term the resulting model extended by semantic information, for instance concerning windows and cars, ‘Virtual habitat’. For generating semantic information, terrestrial images and derived 3D models can be used as well, e.g., for buildings and trees [24,11].

Pollefeys et al. [22] presented one of the first approaches dealing with SfM for a larger number of images in a general configuration, i.e., without known approximate pose. It employed uncalibrated images, i.e., images for which the intrinsic camera parameters such as principal distance (focal length) and principal point are not known. This makes the approach very flexible, yet, on the other hand, reliant on sufficient 3D structure in the scene for the determination of intrinsic parameters.

With the five point algorithm [19], it became feasible to directly compute SfM from calibrated images, i.e., for which the intrinsic parameters are known.

Pollefeys et al. [20] have used it to build a system that was employed for reconstructing 3D structure from more than one hundred thousand images.

Commonly, image overlap is either known implicitly in the form of the order in a sequence, or explicitly, e.g., from an aerial flight plan. Schaffalitzky and Zisserman [25] presented one of the first approaches which automatically determined the image overlap in image sets.

This has led to methods for very large image collections, the so called ‘Community Photo Collections’ – CPC [6] on the Internet. These techniques mostly use information from the Exif (Exchangeable image file format) tags of the images to derive approximate intrinsic camera parameters and thus conduct calibrated SfM. Agarwal et al. [1] have approached the challenge of CPC with a large cloud of computers. Yet, ‘Building Rome on a Cloudless Day’ [5] has dealt with millions of images, for which the only thing known to start with is a tag linking them to a place / city such as Rome. It was shown that the images can be organized in terms of visual similarity. This is used for 3D reconstruction of parts with many images. Everything is computed in one day on one standard computer, albeit with several powerful GPUs – Graphical Processing Units.

While the above work is impressive, one has to note that it is based on certain characteristics of the data and a couple of assumptions which make it tractable:

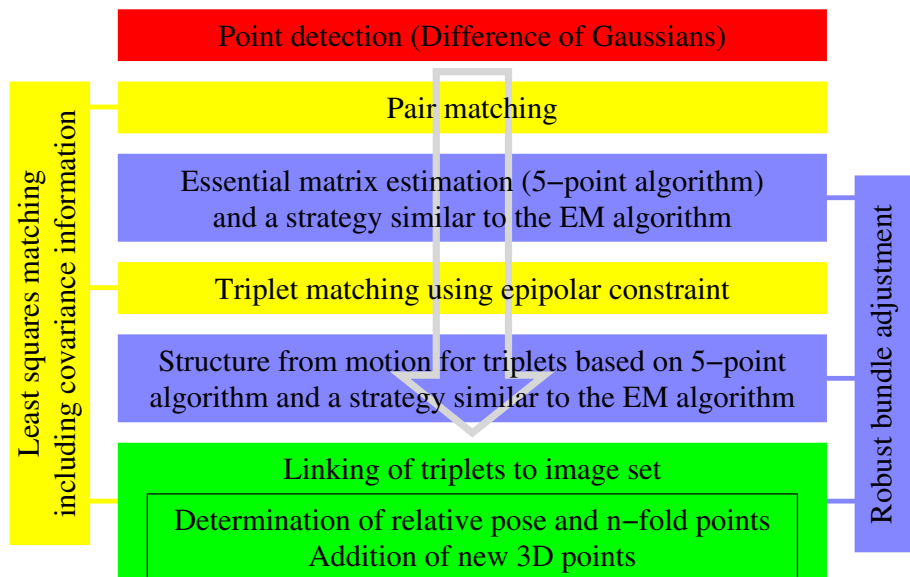
- Many images at tourist attractions are taken from nearly the same spot and thus look alike, i.e., many similar images can be found even for extremely downsampled versions of the images. Frahm et al. [5] use the GIST operator on  $4 \times 4$  images, i.e., very little information on texture and color is available.
- The goal is to reconstruct the obvious 3D structure, leading to impressive 3D reconstructions of highlights, such as the Colosseum in Rome. Yet, there might be images, possibly with wider baselines, that could be used to extend the geometrical coverage or even to link the tourist attractions. This is not considered, as it would mean a detailed comparison of many more images.

A preliminary version of our work, comprising also absolute pose estimation, has been published earlier [2]. It focuses on image sets with possibly very large baselines. For the registration of these images, we have to either supply the sequence of images, or sets of overlapping triplets.

The basis of our work (Figure 1) are points with scale and rotation detected based on Difference of Gaussians (Section 2). We start by removing unlikely matches by cross correlation with a very low threshold. Matches are refined by least squares matching [7] using an affine geometric model. This results in subpixel accurate point positions including covariance information.

The points and their covariance information are employed for SfM from pairs and triplets (Section 3). It is based on a variant of the five point algorithm embedded into RANdom SAMple Consensus – RANSAC [4] using the Geometric Robust Information Criterion – GRIC [27]. A strategy similar to the Expectation Maximization (EM) algorithm is used to extend partial solutions. We employ robust bundle adjustment (Section 4), where we reweight based on residuals (distance between reprojected 3D point and measured 2D point) and, particularly, covariance information for each individual residual.





**Fig. 1.** Structure from Motion based on least squares matching and robust bundle adjustment

Triplets are linked either sequentially or hierarchically to image sets (Section 5). This results in highly precise poses, improved intrinsic parameters, and 3D points including covariance information.

Section 6 presents results for terrestrial images and images acquired from small Unmanned Aircraft Systems (UASs) with a size of less than one meter and a weight of approximately one kilogram. We demonstrate the precision obtained by our approach by means of a loop closing experiment. Wide baseline matching capabilities are shown with results for a combination of terrestrial images and images from a UAS.

The poses and intrinsic parameters are input for Semiglobal Matching – SGM 9 (Section 7) which leads to dense 3D point sets and detailed 2.5D, or 3D surfaces. Finally, results for dense matching with SGM are given. Section 8 concludes the paper with an outlook.

## 2 Point Detection and Matching

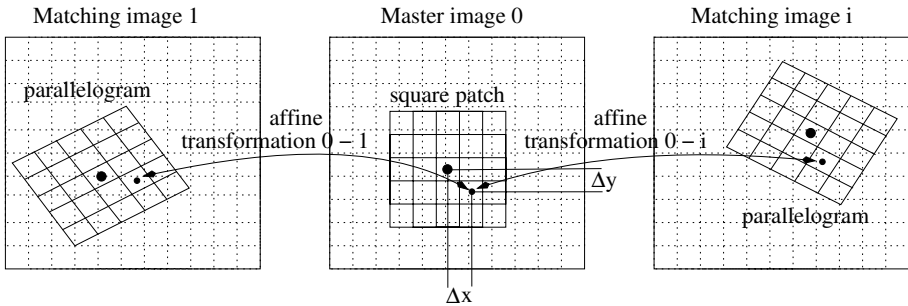
The basis for our approach are points based on Differences of Gaussians (DOG) as proposed by Lowe 15 and implemented in SiftGPU 29. As we want to deal with situations with very low contrast, such as weak structures on facades, we employ a very low threshold.

We start with image pairs. The point centers as well as their estimated scale and rotation are employed to cut out image patches of size  $13 \times 13$  pixels from the

images. These patches are correlated by means of (normalized) cross correlation. For all best matches for points in the master image, which exceed a low threshold of 0.5, we compute a histogram of the rotation differences. The histogram is smoothed and its mode determined. As the mode of the histogram was found to rather reliably describe the in-image-plane rotation between image pairs, we use it for normalization: We cut out unrotated patches (though with individual scale) in one image and rotate all patches in the second image according to the difference of rotation as given by the mode of the histogram.

Cross correlation between patches is computed again and the same low threshold of 0.5 is used. Yet, this time the best matches for all points in the master image exceeding the threshold are subject to least squares matching [7]: The sum of the squared intensity differences between patches around the points is minimized by varying the parameters for a geometric and a radiometric transformation between the patches.

We use an affine geometric model with six parameters  $(a_0^i, \dots, b_2^i)$  describing the translation in x- and y-direction as well as two rotations and two scales. Given a square patch in master image 0, this leads to a parallelogram in the matching images (Figure 2). While the general model for a linear mapping between image patches is a homography, we found that the eight parameters of a homography usually cannot be reliably determined for small patches. Small patches are a must, though, because the region around a point in the scene does not have to be planar and the farther one goes from a point, the higher becomes the likelihood for discontinuities and occlusion.



**Fig. 2.** Least squares matching is based on an affine geometric model. Individual pixels (small dots) of image patches around subpixel precise points (large dots) are transformed based on the affine model. Given a square patch in the master image this leads to parallelograms in the matching images.

The pixel raster of the patch in the master image is defined by  $\Delta x$  and  $\Delta y$  as well as the indices  $j$  and  $k$  ( $-N \leq j \leq N$  and  $-N \leq k \leq N$  with  $N = 6$ ).  $\Delta x$  and  $\Delta y$  depend on the scale known from point detection. The coordinates of the pixels in the master image 0 and the matching image  $i$  are described by

$$\begin{aligned}
 x_{jk}^0 &= x^0 + j\Delta x \\
 x_{jk}^0 &= y^0 + k\Delta y \\
 x_{jk}^i &= x^i + a_0^i + a_1^i j\Delta x + a_2^i k\Delta y \\
 y_{jk}^i &= y^i + b_0^i + b_1^i j\Delta x + b_2^i k\Delta y,
 \end{aligned}
 \tag{1}$$

$$\tag{2}$$

with  $x^0, y^0$  and  $x^i, y^i$  denoting the centers of the patches in master image 0 and matching image  $i$ , respectively. We use subpixel coordinates also for  $x^0$  and  $y^0$  to optimally center the patch around the point.

For the subpixel precise point positions, the intensity of the pixels has to be determined by (in our case bilinear) interpolation. Additionally to the six parameters  $a_0^i, \dots, b_2^i$  for the geometry we use bias  $r_0^i$  and contrast  $r_1^i$  for the intensity to radiometrically adapt the patch in matching image  $i$ . This leads to the following residuals  $v_{jk}^i$  for least squares adjustment ( $I^0()$  and  $I^i()$  denote the intensity function in master image 0 and matching image  $i$ , respectively):

$$v_{jk}^i = I^0(x_{jk}^0, y_{jk}^0) - [r_0^i + r_1^i I^i(x_{jk}^i, y_{jk}^i)]
 \tag{3}$$

The goal of least squares matching is to estimate affine parameters  $a_0^i, \dots, b_2^i$  and radiometric parameters  $r_0^i, r_1^i$  minimizing the sum of all squared residuals

$$\sum_{j=-N}^N \sum_{k=-N}^N [v_{jk}^i]^2.
 \tag{4}$$

Equation (4) is linear with respect to the radiometric parameters  $r_0^i$  and  $r_1^i$ . It is nonlinear in terms of the geometric parameters, because  $I^i()$  is nonlinear in general. As there is no closed-form solution, first order Taylor expansion is employed to linearize Equation (4) based on initial values for the parameters. We assume no translation ( $a_0^i = b_0^i = 0$ ), a similar intensity ( $r_0^i = 0$  and  $r_1^i = 1$ ) and take the known scale difference and rotation into account for  $a_1^i, a_2^i, b_1^i$  and  $b_2^i$ . Setting the derivative to zero, one obtains a linear system

$$\mathbf{A}\beta = \mathbf{y}.
 \tag{5}$$

Matrix  $\mathbf{A}$  consists of the Jacobian of the intensity function in the matching image  $i$  with respect to the unknown geometric and radiometric parameters concatenated in vector  $\beta$ . Vector  $\mathbf{y}$  comprises the negative measurement errors.

While the linear system (5) can be solved directly, we employ the normal equations

$$\mathbf{N}\beta = (\mathbf{A}^T \mathbf{A})\beta = \mathbf{A}^T \mathbf{y}
 \tag{6}$$

and compute  $\beta = \mathbf{N}^{-1} \mathbf{A}^T \mathbf{y}$ . By this means we obtain  $\mathbf{C} = \mathbf{N}^{-1}$ , i.e., the relative covariance matrix for the unknown parameters. Because the problem is nonlinear, the solution is obtained iteratively. For optimization we use the Levenberg Marquardt algorithm.

The criteria for a valid match obtained by least squares matching are that the cross correlation value is larger than 0.8 as well as that the estimated variance

for the shift is below 0.1 pixels. For the latter, one has to consider that from our experience the estimated variance is always highly optimistic. Cross correlation is known to be not a good descriptor for stronger geometrical distortions. Though, it was found to be very useful if the geometrical distortions are small [18], which is the case after least squares matching.

For more than two images, we link least squares matching for pairs. The image in which the patch is closest to the image center is used as master, as this improves the chance for a frontal view. The patch in the master image is geometrically kept as square and the affine transformations relative to the other images are estimated (Figure 2).

The solutions are linked by substituting  $I^0()$  in equation (3) by the average intensity in all images. To account for different average intensities and contrasts of the patches, we take the estimated radiometric parameters  $r_0^i$  and  $r_1^i$  for each patch into account when computing the average. As the problem is nonlinear and solved iteratively, the average intensity changes due to different geometric transformations as well as different radiometric parameters for each iteration.

Output for the accepted matches are the improved coordinates  $x^i + a_0^i$  and  $y^i + b_0^i$  as well as their relative covariance information. The latter can improve SfM estimation particularly for stronger in-image-plane rotations [17].

While least squares matching entails more effort than just using the point centers of the SIFT points, we found that the relative coordinates obtained are more precise. This is probably due to the fact, that we look for optimum matches. This reduces the influence of geometrical deformations, partial occlusions, and noise, which influence point centers when they are estimated independently.

### 3 Two and Three View Geometry

In the remainder of this paper we assume that we have at least an approximate knowledge of the intrinsic parameters. We also implemented an uncalibrated approach in the spirit of Pollefeys et al. [21]. Yet, we found it to be only reliable if sufficient 3D structure is present. Only then, the intrinsic parameters can be reliably determined.

Triplets are the basic geometric building block of our approach due to the following reasons:

- Opposed to pairs where points can only be checked in one dimension by means of their distance from their respective epipolar lines, triplets allow for an unambiguous geometric check of points. This does not only lead to much more reliable points, but also to improved, more reliable information for the cameras.
- Triplets can be directly linked into larger sets by determining their relative pose (translation, rotation, and scale) based on two common images.

Because the combinatorics is worse for triplets than for pairs, we start with pairs and determine essential matrices and thus epipolar lines for them. For known intrinsic parameters, the relative pose of the image pair is determined directly, i.e., with no need for approximate values, by means of the five point algorithm [19].

As usually only a possibly small part of the matched point pairs is actually correct, we employ RANSAC in conjunction with GRIC [27]. The latter means, that instead of counting the number of inliers, we attribute a constant penalty to outliers and values proportional to their squared residuals  $v^2$  to inliers. A threshold is used to define where the transition from inliers to outliers occurs. While in RANSAC the number of inliers is maximized, GRIC aims at a minimum corresponding to many points with small residuals. By means of GRIC one can distinguish between solutions with a low precision, but more points, and highly precise solutions, with possibly less points, but smaller residuals, which are more likely correct.

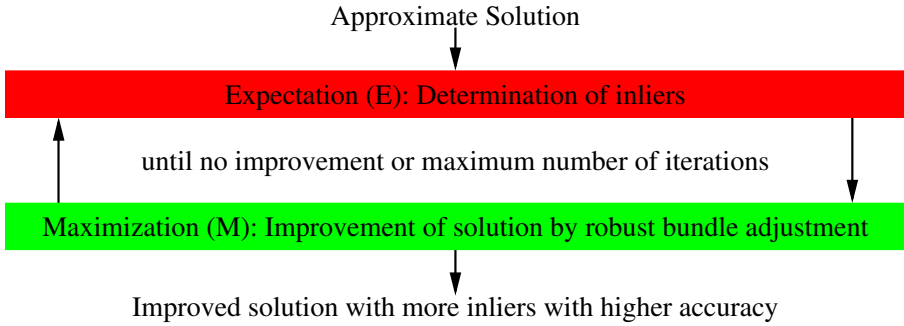
The above combination of RANSAC and GRIC works well for more or less well behaved scenes. Yet, we found that for complex scenes, e.g., involving many very similar points, the above combination is not sufficient to tell good from bad solutions. This happens, e.g., for window corners on facades of buildings, possibly in conjunction with camera movements which conspire with repetitions on the facade. Inspired by Chum et al. [3], we compute a maximum likelihood, i.e., robust bundle adjustment, solution for the best of every couple of hundred RANSAC iterations. Eventually, the bundle solution which leads to the lowest GRIC value is taken as the final result.

But even this gives only a partial solution to the problem. While RANSAC produces a solution from only inliers with a certain probability, it is not guaranteed, that this solution is accurate. Even worse, inaccurate solutions can also be not representative for all, or even the majority of the inliers. E.g., consider a larger image and RANSAC selecting in one sample only inliers from the center of the image. While the geometric solution (of the five point algorithm) will be correct, it will not be precise enough to find also the correct matches closer to the margin of the image. A way to counteract this is to force RANSAC only to use points with a certain minimum distance. Though, this is problematic, because in certain cases there might be just correct matches in the center of the image.

We have devised a strategy similar to the EM algorithm (Figure 3) which employs robust bundle adjustment (cf. next Section) to mitigate the above problem. We robustly bundle adjust the initial direct solution using the inliers determined by RANSAC. The obtained, geometrically improved, solution is employed to compute new inliers based on GRIC. This is iterated until either a predefined number of iterations (here 5) is reached, or no significant improvement in terms of GRIC is obtained.

The above procedure is used for pairs and triplets. For the latter, we employ the result for image pairs to restrict the search space via epipolar lines derived from the essential matrices. This strongly reduces the number of hypotheses for image triplets.

For the geometric computation of triplets, we use one image as master and compute translation and rotation towards the other two images via the five point algorithm for five conjugate points in the three images. This fixes all but one parameter, namely the relative base length between the two pairs. At the moment we assume that we only work with images with a significant base between them.



**Fig. 3.** Strategy similar to the Expectation Maximization (EM) algorithm

While this is a limitation of our approach, we note that there is only a problem with the infinite homography, not with homographies for real planes. Particularly, we triangulate the five points in both pairs and compute the distance from the master image. The ratio of the distances in both pairs is proportional to the ratio of the base lengths. To make the computation robust, we employ the median value of the five ratios computed for the five conjugate points.

## 4 Robust Bundle Adjustment

While bundle adjustment [28] has not been seen as crucial for early approaches on multi view geometry, since a couple of years it is acknowledged that it is useful and even necessary for large image sets.

This is demonstrated by recent work on generalized preconditioners [13]. They allow for an efficient use of conjugate gradient based solutions for bundle adjustment for very large systems also for the general configurations encountered when collecting data from the ground or in CPC.

Our work goes into another direction, namely robustifying bundle adjustment by means of reweighting. I.e., least squares are generalized in the form of an M-estimator [12]. The particular contribution is, that we compute an estimate for the variance of each individual residual and use this for reweighting when implementing the M-estimator.

The estimation of individual variances for the residuals is costly in terms of computation per iteration. Yet, we found that at least for systems with a limited number of images, i.e., tens of images, it is actually faster in the aggregated run time, because much fewer iterations are needed. What is more, one usually obtains a more precise solution consisting of more points.

Following Jian et al. [13], we define  $\mathbf{P} = \{P_i; i = 1, \dots, M\}$  as the camera parameters,  $\mathbf{X} = \{X_j; j = 1, \dots, N\}$  as the 3D points, and  $\mathbf{x} = \{x_k; k = 1, \dots, K\}$  as the measurement of 3D point  $X_j$  in camera  $P_i$ . Function  $f_k(P_i, X_j)$  projects a 3D point to an image. By

$$v_k = f_k(P_i, X_j) - x_k$$

we define the residual between the projected 3D point and the measured image point. The goal of bundle adjustment is to reduce the sum of the squared residuals

$$\sum_{k=1}^K [v_k]^2 . \tag{7}$$

Equation (7) is nonlinear. It can be linearized by means of first order Taylor expansion, assuming that appropriate initial estimates for the camera parameters  $P_i$  and the 3D points  $X_j$  are available:

$$\sum_{k=1}^K [f_k(P_i, X_j) + \frac{\partial f_k(P_i, X_j)}{\partial P_i} dP_i + \frac{\partial f_k(P_i, X_j)}{\partial X_j} dX_j - x_k]^2 . \tag{8}$$

As above for least squares matching, a linear solution (5) can be obtained by setting the derivatives in (8) to zero. The system consists of a sparse matrix  $\mathbf{A}$  made up of the Jacobian of the measurements with respect to cameras and 3D points, the vector  $\beta$  concatenating the parameters of cameras and 3D points, and finally, the vector  $\mathbf{y}$  consisting of the negative measurement errors.

While (5) can be solved directly, we solve the normal equations (6). By this means we can introduce the estimated accuracy of the measured image points as derived by least squares matching in Section 2 in the form of a weight matrix. Particularly, we employ as weight the inverse of the relative covariance matrix of the measurements  $\mathbf{C}$ , leading to

$$\mathbf{N}\beta = (\mathbf{A}^\top \mathbf{C}^{-1} \mathbf{A})\beta = \mathbf{A}^\top \mathbf{C}^{-1} \mathbf{y} . \tag{9}$$

For optimizing the solution, we again use the Levenberg Marquardt algorithm. Please note, that  $\mathbf{C}$  is a positive definite block diagonal matrix consisting of  $2 \times 2$  blocks describing the variance of the measured points in  $x$ - and  $y$ -direction as well as their  $x$ - $y$  covariance.

In the M-estimator, we reweight  $\mathbf{C}$  by

$$w = \sqrt{2 + \bar{v}^2} ,$$

with  $\bar{v} = v/\sigma_v$ . I.e., the residual is divided by its standard deviation. While usually a common variance is used, we compute an estimate of the covariance of the individual residuals  $C_v$  as follows:

$$\mathbf{C}_v = \mathbf{C} - \mathbf{A}(\mathbf{A}^\top \mathbf{C}^{-1} \mathbf{A})^{-1} \mathbf{A}^\top = \mathbf{C} - \mathbf{A} \mathbf{N}^{-1} \mathbf{A}^\top \tag{10}$$

For an efficient solution, we employ the Schur complement and split up the design matrix in a part for 3D points  $\mathbf{A}_X$  and a part for the cameras  $\mathbf{A}_C$ . This results in the following (symmetric) matrix  $\mathbf{N}$  and its inverse  $\mathbf{M}$

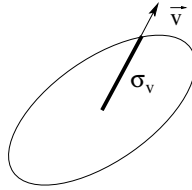
$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_{XX} & \mathbf{N}_{XC} \\ \mathbf{N}_{XC}^\top & \mathbf{N}_{CC} \end{bmatrix} \quad \text{and} \quad \mathbf{M} = \mathbf{N}^{-1} = \begin{bmatrix} \mathbf{M}_{XX} & \mathbf{M}_{XC} \\ \mathbf{M}_{XC}^\top & \mathbf{M}_{CC} \end{bmatrix} .$$

We solve for  $\mathbf{M}_{CC} = (N_{CC} - \mathbf{N}_{XC}^T \mathbf{N}_{XX} \mathbf{N}_{XC})^{-1}$ , i.e., the inverse for the cameras, at the core of the bundle adjustment. The computation of  $\mathbf{M}_{XX} = \mathbf{N}_{XX}^{-1} + \mathbf{N}_{XX}^{-1} \mathbf{N}_{XC} \mathbf{M}_{CC} \mathbf{N}_{XC}^T \mathbf{N}_{XX}^{-1}$  can be done very efficiently, as it only involves the inversion of  $3 \times 3$  matrices in the block diagonal matrix  $\mathbf{N}_{XX}$  and multiplications with  $3 \times 6$  and  $6 \times 6$  matrices. The covariance between points and cameras  $\mathbf{M}_{XC}$  is for most applications not needed and, thus, not calculated. From  $\mathbf{N} \cdot \mathbf{M} = \mathbf{I}$  (with  $\mathbf{I}$  the unit matrix) one can derive

$$\mathbf{M}_{XC} = \mathbf{N}_{XX}^{-1} \mathbf{N}_{XC} \mathbf{M}_{CC} ,$$

giving the full matrix  $\mathbf{M} = \mathbf{N}^{-1}$  needed to solve Equation (10).

As the measurements are 2D image coordinates, the covariance information for residuals corresponds to 2D ellipses. Thus,  $\bar{v} = v/\sigma_v$  is computed as ratio of the length of the residual vector and the standard deviation of the residual in the direction of the residual  $\sigma_v$  as shown in Figure 4.  $\bar{v}$  is employed to reweight the  $2 \times 2$  block in matrix  $\mathbf{C}$  corresponding to the residual.



**Fig. 4.** Error ellipse for residual, direction of the residual and the standard deviation in the direction of the residual  $\sigma_v$

## 5 Structure from Motion for Image Sets

We link image sets based on camera information for two common images. We start by linking triplets, but depending on the strategy (cf. below), also sets are linked to sets.

For obtaining approximate values, we first relate the camera information for an image in one set, i.e., the master set, to the camera information for the same image in the other set, i.e., the slave set. As we assume that we know the intrinsic parameters, we can translate and rotate the slave into the master set. The remaining unknown is scale. It is derived from the camera parameters for a second common image, for which in both images the distance to the first common camera is computed. The ratio of the distances gives the ratio in scale of the two coordinate systems. With the obtained approximate values for translation, rotation, and scale, we transform all camera parameters and 3D points from the slave into the master set.

Additionally, we transform also points from the master into the slave set, to obtain more than twofold, i.e.,  $n$ -fold points<sup>1</sup>. The higher  $n$ , the more geometrically stable the solution becomes. For computing  $n$ -fold points, we first

<sup>1</sup> The terms twofold, threefold, and  $n$ -fold point are used for expressing that the projection of a 3D scene point is detected in two, three, or  $n$  images.



note, that it is not useful to compare points in 3D space, because its metric is in general not well defined. Thus, we conduct the comparison in image space. Particularly, we employ trifocal tensors computed from the camera matrices [8] of the slave set and project points from the two common images of the master set into the third, etc., image of the slave set. There, multi-image least squares matching (Section 2) is conducted leading to  $n$ -fold points. Finally, we compute a robust bundle adjustment (Section 4) based on the approximate values for translation, rotation, and scale, as well as the  $n$ -fold points.

This gives an improved solution for the overlapping part of the combined set. Yet, novel points in the slave set are still missing. Therefore, we compare for the two common images the image coordinates from the slave set with the image coordinates in the master set. Only when there is no nearby point found in image space as implemented by dilation with a radius of two pixels, the corresponding 3D point is introduced. Eventually, again a robust bundle adjustment is computed, this time also including the estimation of improved intrinsic parameters.

We note that the above procedure tracks a point only as long it is visible. While this means that points which are occluded in a frame are lost and possibly re-introduced, we found that this is superior to projecting 3D points into the images. The problem with the latter is, that if one goes around an object, repeating structures, possibly even on the backside, can by chance be at the same location and match very well. As these points are wrong, they can introduce a serious bias in the estimation.

For linking sets, we have implemented a

- sequential strategy and a
- hierarchical strategy.

The sequential strategy is very simple: We just link one triplet after the other to the set with an overlap of always two images. The basic problem with this simple strategy is, that at least for wide baseline sets we found it is necessary that we conduct a robust bundle adjustment each time we add a triplet. This makes the strategy computationally very intensive.

On the other hand, in the hierarchical strategy, sub-sets are grown in parallel and linked one by one (Table 1). As we need two common images, this means that we can extend the set by  $2i - 2$  images. Starting with 3, we obtain sets with 4, 6, 10, 18, 34, etc. images. This is obviously much more efficient as it entails much fewer robust bundle solutions.

It is less obvious, though, that the hierarchical strategy is also very useful in terms of robust bundle adjustment, particularly for large sets. For robust reweighting (Section 4), it is important, that the variances of the residuals are comparable. If this is not the case, e.g., when linking a large set with multiple overlap and high internal precision with a small set and thus with low precision, there is a strong tendency, that a considerable number of the weaker, but correct points of the smaller set will be thrown out. All this is avoided by hierarchical linking, where sets of approximately the same size and, thus, precision are linked.

**Table 1.** Hierarchical linking eight image triplets for ten images

1 2 3	2 3 4	3 4 5	4 5 6	5 6 7	6 7 8	7 8 9	8 9 10
1 [2 3] 4		3 [4 5] 6		5 [6 7] 8		7 [8 9] 10	
	1 2 [3 4] 5 6				5 6 [7 8] 9 10		
			1 2 3 4 [5 6] 7 8 9 10				

This was demonstrated by Mayer [16] for a loop of ninety images taken inside the Zwinger, Dresden, Germany. Hierarchical linking has been seven times faster. More important, it produced not only 32,783 compared to 28,582 points for sequential linking, but also many more many-fold points.

## 6 Results of Structure from Motion

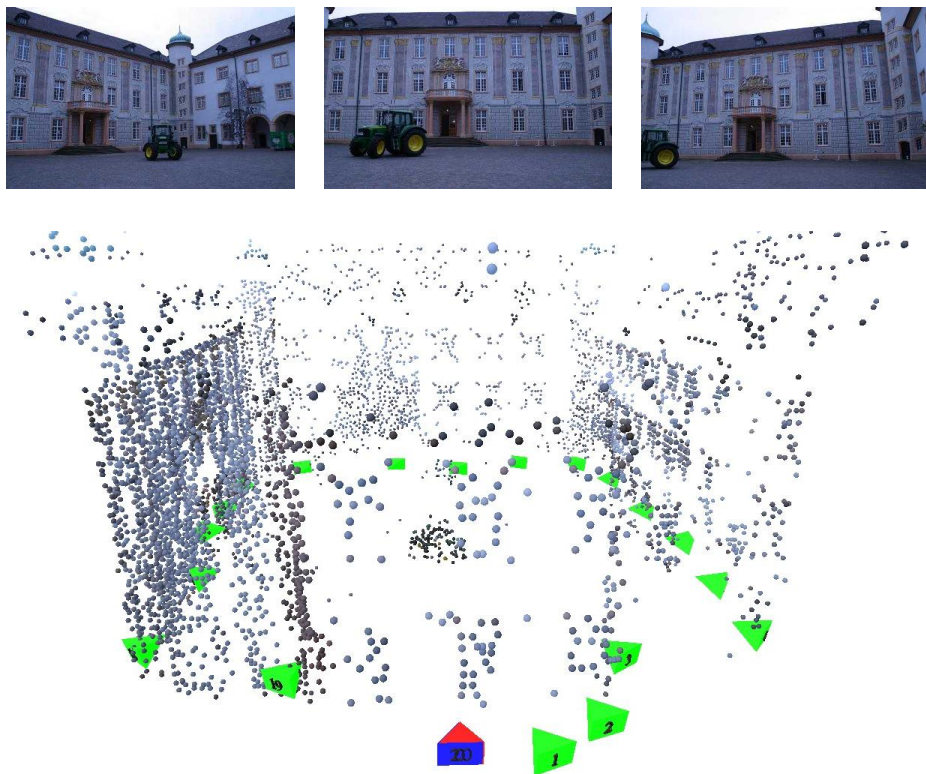
All experiments reported in this section have been conducted using the sequential strategy and the same parameters.

The sequence castle-R20 of Ettlingen castle in Germany consists of twenty images [26]. Some of them are shown at the top of Figure 5. Our SfM approach results in an estimated average back-projection error  $\sigma_0$  of 0.14 pixels. For demonstrating the high precision of our results, we conducted a loop closing experiment, i.e., we took the last image of the twenty images sequence to be the same as the first image. SfM was conducted without closing the sequence. This means that the differences between the camera parameters for the first and the last image, which should be the same, give an indication of the precision obtained.

Firstly, we note that Figure 5 visually shows, that the differences are small. Table 2 gives a quantitative evaluation. The upper part shows the translation error. It is in the range of 0.1 % of the maximum distance between the camera centers. In terms of an absolute distance this means about 4 centimeters. The absolute angular error after twenty images is only  $0.14^\circ$ . This means that we obtained a relative angular error per image of  $0.007^\circ$ , demonstrating the high precision achieved.

The top of Figure 6 shows three pairs of near infrared images of size  $1392 \times 1040$  pixel of a sequence of 400 images taken by a mobile mapping system. The pairs have a small overlap due to a diverging imaging configuration and the images a limited quality due to the near infrared. In spite of this and even though the images were not explicitly treated as pairs in SfM, but as sequence, the local geometry of the pairs could be estimated very well. This is mainly due to robustly tracking points over many frames resulting in highly precise many-fold points and camera poses.

The third example is based on images acquired for a village in southern Germany by a small UAS. In one experiment, a building has been captured by terrestrial images which have been linked via ascending images (center of Figure 7 bottom) to a flight line above the village. In spite of the partially strong wide baseline geometry (Figure 7, center row), we could still compute valid and precise camera poses and 3D points.



**Fig. 5.** Top: Images four, seven, and eight of image sequence castle-R20 of Ettlingen castle in Germany, with twenty images [26]. Bottom: Result for SfM ( $\sigma_0 = 0.14$  pixels). Cameras are given as pyramids and points are colored from the images. For the loop closing experiment, the first and the last image of the sequence were taken to be the same, depicted in red and blue with numbers 0 and 20. The overlap of the latter demonstrates the high quality of the reconstruction.

## 7 Dense Reconstruction

For dense reconstruction we employ Semiglobal Matching – SGM [9]. It is based on

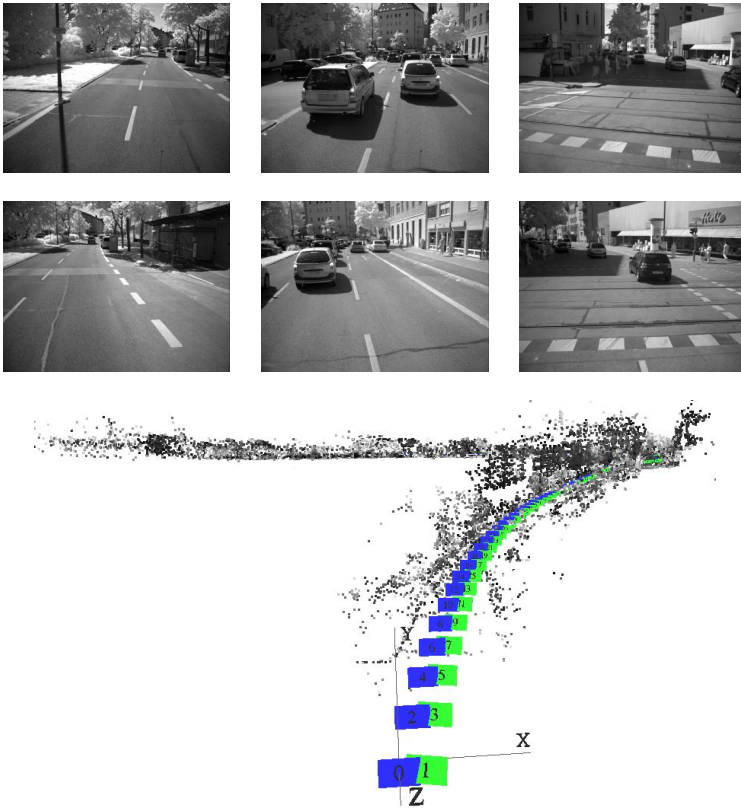
- mutual information (MI) or the Census filter for cost computation and
- the substitution of a 2D smoothness term by a combination of 1D constraints (semiglobal).

The mutual information  $mi_{I_1, I_2}$  is the sum of the entropies in the two images to be matched  $h_{I_1}(i)$  and  $h_{I_2}(k)$  minus their joint entropy  $h_{I_1, I_2}(i, k)$

$$mi_{I_1, I_2} = h_{I_1}(i) + h_{I_2}(k) - h_{I_1, I_2}(i, k) \quad . \quad (11)$$

**Table 2.** Evaluation for castle-R20 in terms of loop closing error – Top: Translation in terms of maximum distance of projection centers as well as in absolute distance; Bottom: Absolute angular error (after twenty images) and relative angular error (per image)

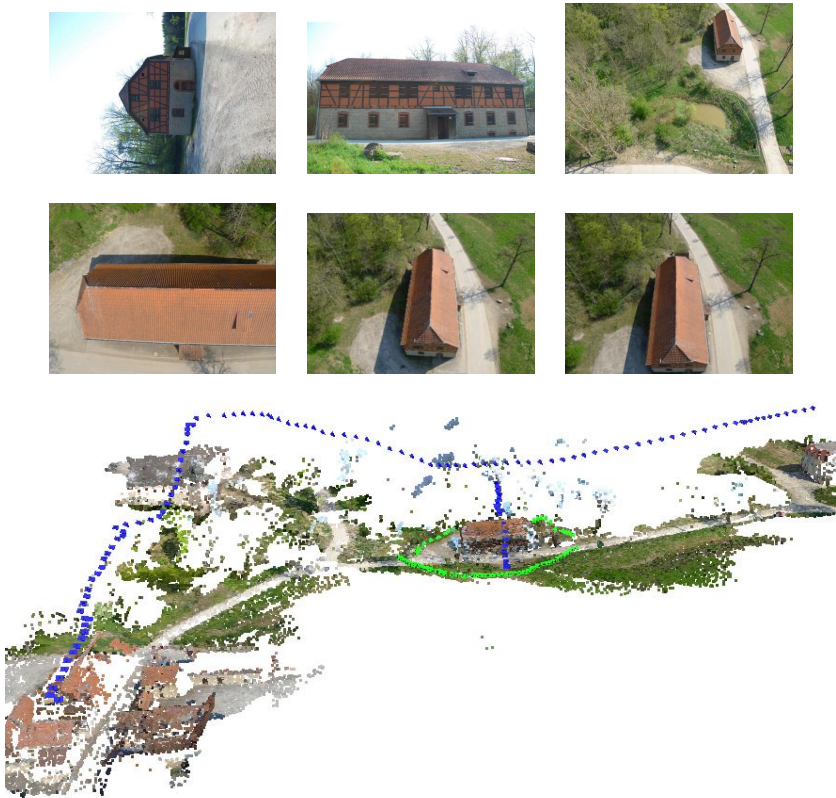
Translation	x	y	z
% of maximum distance	0.124	-0.011	-0.053
absolute distance [m]	0.041	-0.004	-0.017
Absolute angular error			0.1398°
Relative angular error per image			0.0070°



**Fig. 6.** Top: Image pairs of an infrared sequence of 400 images taken by a mobile mapping system given in the form top / bottom from left to right: Pairs 4 / 5, 118 / 119, and 180 / 181. Bottom: Result of SfM. Points are given with the color taken from the images and camera positions and orientations are marked by colored pyramids.

This leads to the following matching cost ( $f_D$  transforms the matching image  $I_m$  with an initial disparity image  $D$ )

$$C_{MI}(\mathbf{p}, d) = -mi_{I_b, f_D(I_m)}(I_{b\mathbf{p}}, I_{m\mathbf{q}}) \quad , \quad (12)$$

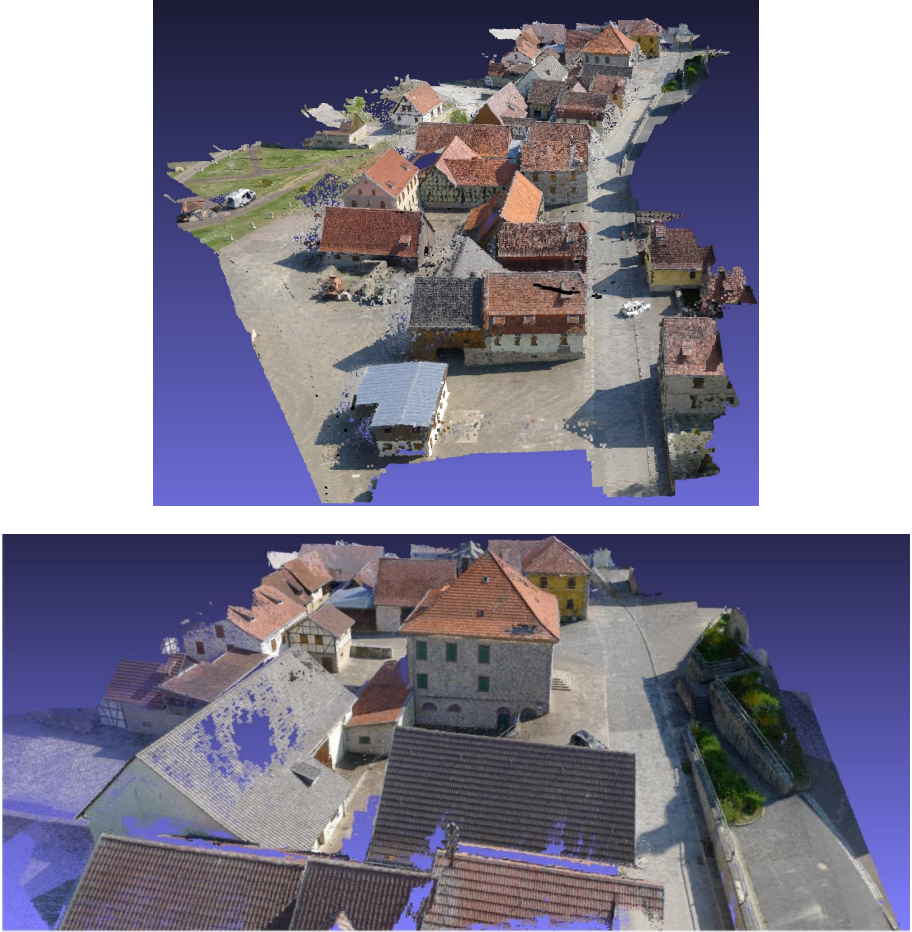


**Fig. 7.** Top: Images of a German village taken from the ground and from an ascending UAS. Please note the wide baselines between the left and the other two images of the triplet shown on the second row. Bottom: Result for SfM estimation. Cameras are given as pyramids and points are colored from the images. For the building in the center terrestrial images have been linked to the flight line above via ascending images.

where  $\mathbf{q}$  is the pixel in the matching image  $I_m$  corresponding to the pixel  $\mathbf{p}$  in the reference image  $I_b$  and the disparity  $d$ .

In essence, MI gives the conditional probability distribution for the intensities in the matching image given an intensity in the reference image without resorting to a parametric model. Thus, MI can compensate a large class of global radiometric differences. Though, one has to note that the conditional probability is computed for the whole image which can be a problem for local radiometric changes, e.g., if materials with very different reflection characteristics exist in the scene or lighting conditions change.

The Census filter was found by Hirschmüller and Scharstein [10] to be the most robust variant for matching cost computation. It defines a bit string with each bit corresponding to a pixel in the local neighborhood of a given pixel. A bit is set if the intensity is lower than that of the given pixel. Census thus encodes

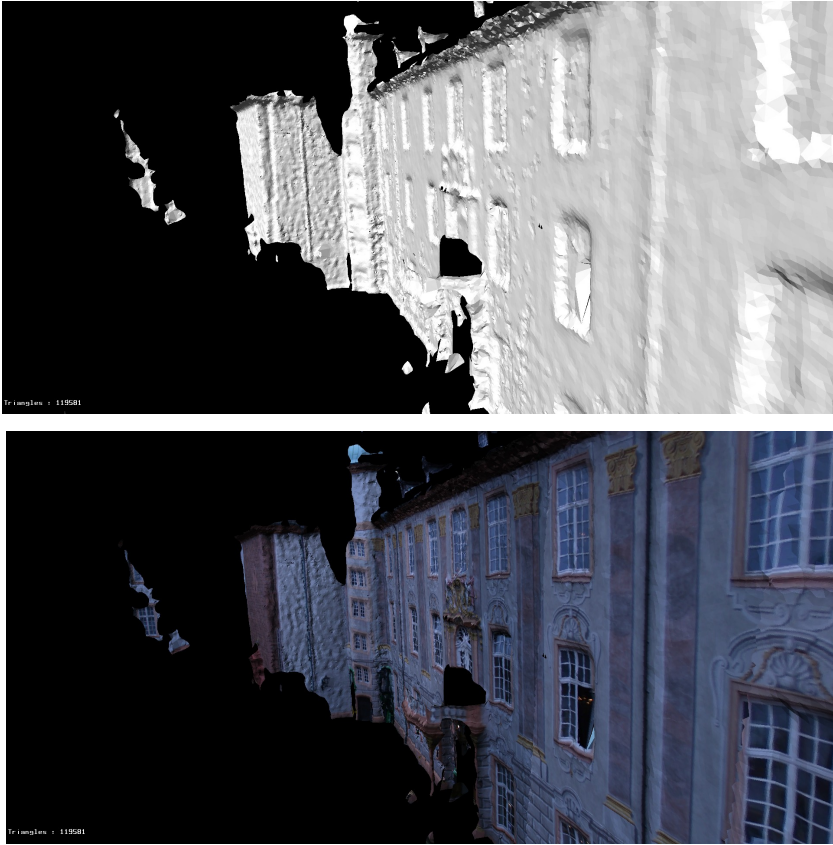


**Fig. 8.** Top: Dense 3D points generated by SGM. Bottom: Part

the spatial neighborhood structure. A  $7 \times 9$  neighborhood can be encoded in a 64 bit integer. Matching is conducted via computing the Hamming distance between corresponding bit strings.

The smoothness term punishes changes of neighboring disparities (operator  $T[\cdot]$  is 1 if its argument is true and 0 otherwise):

$$\begin{aligned}
 E(D) = \sum_{\mathbf{p}} \left( C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_1 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] \right. \\
 \left. + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_2 T[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right) \quad (13)
 \end{aligned}$$



**Fig. 9.** Result for dense 3D surface mesh reconstruction using SGM of parts of image sequence castle-R20 (Figure 5) – shaded (top) and textured (bottom)

- The first term consists of pixel matching costs for all disparities of  $D$ .
- The second term adds a constant penalty  $P_1$  for all pixels  $\mathbf{q}$  from the neighborhood  $N_{\mathbf{p}}$  of  $\mathbf{p}$ , for which the disparity changes only slightly (1 pixel).
- The third term adds a larger constant penalty  $P_2$  for bigger changes of the disparities. Because it is independent of the size of the disparities, it preserves discontinuities.
- As discontinuities in disparity are often visible as intensity changes,  $P_2$  is calculated depending on the intensity gradient in the reference image (with  $P_2 \geq P_1$ ).

In 2D, global minimization is NP hard for many discontinuity preserving energies  $E(D)$ . In 1D, minimization can be done in polynomial time via dynamical programming, which is usually applied within image lines. Unfortunately, because the solutions for neighboring lines are computed independently, this can lead to streaking.

For the semiglobal solution, 1D matching costs are computed in different, (practically 8) directions which are aggregated without weighting. In the reference image, straight lines are employed, which are deformed in the matching image.

By computing  $D$  for exchanged reference and matching image one can infer occlusions or matching errors by means of a consistency check. If more than one pair with the same reference image is matched, the consistency check is conducted for all pairs only once.

With the above methodology, dense disparities can be computed. By using the camera parameters all points can be projected into 3D leading to dense 3D point clouds. While the original work of Hirschmüller [9] has shown how to derive 2.5D surface models, work on the derivation of a 3D surface by means of triangulation of the 3D points dealing also with outliers has been started only recently.

For parts of the village for which camera poses and 3D point clouds have been estimated (Section 6, Figure 7), SGM was used to compute dense 3D points from several pairs. Figure 8 gives an impression of the very high point density and quality obtained.

Finally, Figure 9 shows first results for dense 3D surface mesh reconstruction using SGM. Particularly the shaded visualization shows, that the indentations of the windows could be determined reliably.

## 8 Conclusions and Outlook

In this paper we have presented an approach for dense reconstruction from wide baseline image sets. As key characteristics it aims at a high precision in every step of the approach from least squares matching to robust bundle adjustment. Particularly for the latter, we take into account the estimated covariance for the residuals, leading to more precise solutions with more points in less time.

Even though we have demonstrated that we can compute SfM for larger scenes consisting of hundreds of images with wide baselines, there are still a couple of shortcomings. The most basic is, that we rely on given information concerning image overlap. While Agarwal et al. [1] and Frahm et al. [5] have shown how the problem can be solved in principle, it is still not clear how to deal with wide baselines. The most obvious way is to compare all possible pairs, but for larger sets this seems to be not feasible even using GPUs.

Yet, also for large scenes with small baselines problems exist. One is in the line of thought of our hierarchical approach for linking image sets (Section 5). Particularly, the question is, which parts of the unordered sets should be linked when, i.e., at which level of the hierarchy.

Then, there are problems with objects of the real world with specific characteristics. E.g., some objects have symmetries, such as that front and back look very similar. This is hard for current approaches for unordered sets, where missing matches are usually attributed to unmodeled occlusions. Thus, the questions arises, how much semantic information is needed for a reliable 3D reconstruction? Should ordering information from the camera, e.g., in terms of known acquisition time be used? If location information, e.g., from GPS is available and reliable, it could be used to circumvent the problem.



Finally, there are also a couple of smaller or larger details in our approach which could be solved in a better way. E.g., at the moment we use one standard value for RANSAC / GRIC for pairs and triplets. While this works in nearly all cases, it can be far from optimal as it does not account for the different precisions possible for images of different sizes, distortions, lighting, contrast and scene characteristics (e.g., facade planes versus trees). Here, estimation by means of RECON [23] could give a more general solution.

**Acknowledgment.** We thank the reviewers for their comments, which have helped to make important parts much more explicit.

Parts of the presented work were supported by Bundeswehr Geoinformation Office which is gratefully acknowledged.

## References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a Day. In: Twelfth International Conference on Computer Vision, pp. 72–79 (2009)
2. Bartelsen, J., Mayer, H.: Orientation of Image Sequences Acquired from UAVs and with GPS Cameras. *Surveying and Land Information Science* 70(3), 151–159 (2010)
3. Chum, O., Matas, J., Kittler, J.: Locally Optimized RANSAC. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 236–243. Springer, Heidelberg (2003)
4. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6), 381–395 (1981)
5. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010)
6. Goesele, M., Ackermann, J., Fuhrmann, S., Klowy, R., Langguth, F., Muecke, P., Ritz, M.: Scene Reconstruction from Community Photo Collections. *IEEE Computer* 43(6), 48–53 (2010)
7. Grün, A.: Adaptive Least Squares Correlation: A Powerful Image Matching Technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography* 14(3), 175–187 (1985)
8. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
9. Hirschmüller, H.: Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), 328–341 (2008)
10. Hirschmüller, H., Scharstein, D.: Evaluation of Stereo Matching Costs on Images with Radiometric Differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(9), 1582–1599 (2009)
11. Huang, H., Mayer, H.: Generative Statistical 3D Reconstruction of Unfoliated Trees from Terrestrial Images. *Annals of GIS* 15(2), 97–105 (2009)
12. Huber, P.: *Robust Statistics*. John Wiley & Sons, Inc., New York (1981)

13. Jian, Y.D., Balcan, D., Dellaert, F.: Generalized Subgraph Preconditioners for Large-Scale Bundle Adjustment. In: Thirteenth International Conference on Computer Vision, pp. 295–302 (2011)
14. Leberl, F., Bischof, H., Pock, T., Irschara, A., Kluckner, S.: Aerial Computer Vision for a 3D Virtual Habitat. *IEEE Computer* 43(6), 24–31 (2010)
15. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
16. Mayer, H.: Efficiency and Evaluation of Markerless 3D Reconstruction from Weakly Calibrated Long Wide-Baseline Image Loops. In: 8th Conference on Optical 3-D Measurement Techniques, vol. II, pp. 213–219 (2007)
17. Mayer, H.: Issues for Image Matching in Structure from Motion. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. (37) B3a, pp. 21–26 (2008)
18. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(10), 1615–1630 (2005)
19. Nistér, D.: An Efficient Solution to the Five-Point Relative Pose Problem. In: *Computer Vision and Pattern Recognition*, vol. II, pp. 195–202 (2003)
20. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénus, H., Yang, R., Welch, G., Towles, H.: Detailed Real-Time Urban 3D Reconstruction from Video. *International Journal of Computer Vision* 78(2-3), 143–167 (2008)
21. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J.: Visual Modeling with a Hand-Held Camera. *International Journal of Computer Vision* 59(3), 207–232 (2004)
22. Pollefeys, M., Verbiest, F., Van Gool, L.: Surviving Dominant Planes in Uncalibrated Structure and Motion Recovery. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part II*. LNCS, vol. 2351, pp. 837–851. Springer, Heidelberg (2002)
23. Raguram, R., Frahm, J.M.: RECON: Scale-Adaptive Robust Estimation via Residual Consensus. In: Thirteenth International Conference on Computer Vision, pp. 1299–1306 (2011)
24. Reznik, S., Mayer, H.: Implicit Shape Models, Self Diagnosis, and Model Selection for 3D Facade Interpretation. *Photogrammetrie – Fernerkundung – Geoinformation* 3(08), 187–196 (2008)
25. 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)
26. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: *Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
27. Torr, P.: An Assessment of Information Criteria for Motion Model Selection. In: *Computer Vision and Pattern Recognition*, pp. 47–53 (1997)
28. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle Adjustment – A Modern Synthesis. In: *Workshop on Vision Algorithms in conjunction with ICCV 1999*, pp. 298–372 (1999)
29. Wu, C.: SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT) (2007), [cs.unc.edu/~ccwu/siftgpu](http://cs.unc.edu/~ccwu/siftgpu)

# Data-Driven Manifolds for Outdoor Motion Capture

Gerard Pons-Moll<sup>1</sup>, Laura Leal-Taixé<sup>1</sup>, Juergen Gall<sup>2,3</sup>, and Bodo Rosenhahn<sup>1</sup>

<sup>1</sup> Leibniz University, Hannover, Germany

<sup>2</sup> BIWI, ETH Zurich, Switzerland

<sup>3</sup> MPI for Intelligent Systems, Germany

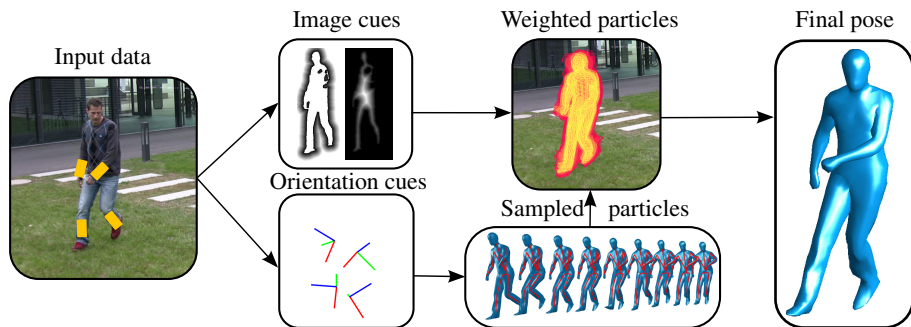
**Abstract.** Human motion capturing (HMC) from multiview image sequences is an extremely difficult problem due to depth and orientation ambiguities and the high dimensionality of the state space. In this paper, we introduce a novel hybrid HMC system that combines video input with sparse inertial sensor input. Employing an annealing particle-based optimization scheme, our idea is to use orientation cues derived from the inertial input to sample particles from the manifold of valid poses. Then, visual cues derived from the video input are used to weight these particles and to iteratively derive the final pose. As our main contribution, we propose an efficient sampling procedure where the particles are derived analytically using inverse kinematics on the orientation cues. Additionally, we introduce a novel sensor noise model to account for uncertainties based on the von Mises-Fisher distribution. Doing so, orientation constraints are naturally fulfilled and the number of needed particles can be kept very small. More generally, our method can be used to sample poses that fulfill arbitrary orientation or positional kinematic constraints. In the experiments, we show that our system can track even highly dynamic motions in an outdoor environment with changing illumination, background clutter, and shadows.

## 1 Introduction

Recovering 3D human motion from 2D video footage is an active field of research [21, 37, 10, 33, 37]. Although extensive work on human motion capturing (HMC) from multiview image sequences has been pursued for decades, there are only few works, e.g. [15], that handle challenging motions in outdoor scenes.

To make tracking feasible in complex scenarios, motion priors are often learned to constrain the search space [18, 29, 30, 32, 37]. On the downside, such priors impose certain assumptions on the motions to be tracked, thus limiting the applicability of the tracker to general human motions. While approaches exist to account for transitions between different types of motion [2, 5, 11], general human motion is highly unpredictable and difficult to be modeled by pre-specified action classes.

Even under the use of strong priors, video HMC is limited by current technology: depth ambiguities, occlusions, changes in illumination, as well as shadows and background clutter are frequent in outdoor scenes and make state-of-the-art algorithms break down. Using many cameras does not resolve the main difficulty in outdoor scenes, namely extracting reliable image features. Strong lighting conditions also rule out the use of depth cameras. Inertial sensors (IMU) do not suffer from such limitations but they are intrusive by nature: at least 17 units must be attached to the body which poses



**Fig. 1.** Orientation cues extracted from inertial sensors are used to efficiently sample valid poses using inverse kinematics. The generated samples are evaluated against image cues in a particle filter framework to yield the final pose.

a problem from bio-mechanical studies and sports sciences. Additionally, IMU's alone fail to measure accurately translational motion and suffer from drift. Therefore, similar to [27][24][35], we argue for a hybrid approach where visual cues are supplemented by orientation cues obtained by a small number of additional inertial sensors. While in [35] only arm motions are considered, the focus in [24] is on indoor motions in a studio environment where the cameras and sensors can be very accurately calibrated and the images are nearly noise- and clutter-free. By contrast, we consider full-body tracking in an outdoor setting where difficult lighting conditions, background clutter, and calibration issues pose additional challenges. The work presented here is an extension of our previous article [27]. Here, we extend it and show more results and more implementation details of the proposed approach.

In this paper, we introduce a novel hybrid tracker that combines video input from four consumer cameras with orientation data from five inertial sensors, see Fig. 1. Within a probabilistic optimization framework, we present several contributions that enable robust tracking in challenging outdoor scenarios. Firstly, we show how the high-dimensional space of all poses can be projected to a lower-dimensional manifold that accounts for kinematic constraints induced by the orientation cues. To this end, we introduce an explicit analytic procedure based on Inverse Kinematics (IK). Secondly, by sampling particles from this low-dimensional manifold the constraints imposed by the orientation cues are implicitly fulfilled. Therefore, only a small number of particles is needed, leading to a significant improvement in efficiency. Thirdly, we show how to integrate a sensor noise model based on the von Mises-Fisher [8] distribution in the optimization scheme to account for uncertainties in the orientation data. In the experiments, we demonstrate that our approach can track even highly dynamic motions in complex outdoor settings with changing illumination, background clutter, and shadows. We can resolve typical tracking errors such as miss-estimated orientations of limbs and swapped legs that often occur in pure video-based trackers. Moreover, we compare it with three different alternative methods to integrate orientation data. Finally, we make the challenging dataset and sample code used in this paper available for scientific use<sup>1</sup>.

<sup>1</sup> <http://www.tnt.uni-hannover.de/~pons/>

## 2 Related Work

For solving the high-dimensional pose optimization problem, many approaches rely on local optimization techniques [4,15,28], where recovery from false local minima is a major issue. Under challenging conditions, global optimization techniques based on particle filters [7,10,38,26] have proved to be more robust against ambiguities in the data. Thus, we build upon the particle-based annealing optimization scheme described in [10]. Here, one drawback is the computational complexity which constitutes a bottleneck when optimizing in high-dimensional pose spaces.

Several approaches show that constraining particles using external pose information sources can reduce ambiguities [1,12,13,16,17,20,34]. For example, [17] uses the known position of an object a human actor is interacting with and [120] use hand detectors to constrain the pose hypotheses. To integrate such constraints into a particle-based framework, several solutions are possible. Firstly, the cost function that weights the particles can be augmented by additional terms that account for the constraints. Although robustness is added, no benefits in efficiency are achieved, since the dimensionality of the search space is not reduced. Secondly, rejection sampling, as used in [17], discards invalid particles that do not fulfill the constraints. Unfortunately, rejection sampling can be very inefficient and does not scale well with the number of constraints as we will show. Thirdly, approaches such as [9,12,19,34] suggest to explicitly generate valid particles by solving an IK problem on detected body parts. While the proposals in [19,34] are tailored to deal with depth ambiguities in monocular imagery, [12] relies on local optimization which is not suited for outdoor scenes as we will show. In the context of particle filters, the von Mises-Fisher distribution has been used as prior distribution for extracting white matter fiber pathways from MRI data [40].

In contrast to previous work, our method can be used to sample particles that fulfill arbitrary kinematic constraints by reducing the dimension of the state space. Furthermore, none of the existing approaches perform a probabilistic optimization in a constrained low-dimensional manifold. We introduce an IK based on the *Paden-Kahan* sub-problems and model rotation noise with the von Mises-Fisher distribution.

## 3 Global Optimization with Sensors

To temporally align and calibrate the input data obtained from a set of uncalibrated and unsynchronized cameras and from a set of orientation sensors, we apply preprocessing steps as explained in Sect. 3.1. Then, we define orientation data within a human motion model (Sect. 3.2) and explain the probabilistic integration of image and orientation cues into a particle-based optimization framework (Sect. 3.3).

### 3.1 Calibration and Synchronization

We recorded several motion sequences of subjects wearing 10 inertial sensors (we used XSens [36]) which we split in two groups of 5: the *tracking sensors* which we use for tracking and the *validation sensors* which we use for evaluation. According to the specifications, the IMU orientation accuracy is around  $2^\circ$  for smooth motions and in

absence of magnetic field. In practice, unfortunately, the error is much higher due to different sources of uncertainty, see Sect. 4.3. The tracking sensors are placed in the back and the lower limbs and the validation sensors are placed on the chest and the upper limbs. An inertial sensor  $s$  measures the orientation of its local coordinate system  $F_s^S$  w.r.t. a fixed global frame of reference  $F^T$ . All sensors derive the same global frame of reference by merging information from a magnetic field sensor, an accelerometer and a rate gyro. The orientation data is given as a stream of rotation matrices  $\mathbf{R}_s^{TS}(t)$  that define the coordinate transform from  $F_s^S$  to  $F^T$ . In the process of calibrating the camera, the global tracking coordinate system  $F^T$  is defined by a calibration cube placed into the recording volume. In order to bring  $F^I$  and  $F^T$  into correspondence, we carefully place the calibration cube such that the axes of  $F^T$  directly correspond to the axes of the known  $F^I$  using a compass. Like this, the orientation data  $\mathbf{R}_s^{IS}(t)$  also directly maps from the local sensor coordinate system  $F_s^S$  to the global tracking coordinate system  $F^T$  and we note  $\mathbf{R}^{TS} := \mathbf{R}^{IS}$ . Note that there might be slight misalignments between the tracking and inertial frame for which we compensate by introducing a sensor noise model, see Sec. 4.3. In this paper, we refer to the sensor orientations by  $\mathbf{R}^{TS}$  and, where appropriate, by using the corresponding quaternion representation  $\mathbf{q}^{TS}$ . Quaternions generalize complex numbers and can be used to represent 3D rotations the same way as complex numbers can be used to represent planar rotations [31]. The video sequences recorded with four off-the-shelf consumer cameras are synchronized by cross correlating the audio signals as proposed in [15]. Finally, we synchronize the IMU's with the cameras using a clapping motion, which can be detected in the audio data as well as in the acceleration data measured by IMU's.

### 3.2 Human Motion Model

We model the motion of a human by a skeletal kinematic chain containing  $N = 25$  joints that are connected by rigid bones. The global position and orientation of the kinematic chain are parameterized by a twist  $\xi_0 \in \mathbb{R}^6$  [22]. A twist is an element of the tangent space of rigid body motions, see [26] for a comprehensive introduction to human body parameterizations. Together with the joint angles  $\Theta := (\theta_1 \dots \theta_N)$ , the configuration of the kinematic chain is fully defined by a  $D=6+N$ -dimensional vector of pose parameters  $\mathbf{x} = (\xi_0, \Theta)$ . We now describe the relative rigid motion matrix  $\mathbf{G}_i$  that expresses the relative transformation introduced by the rotation in the  $i$ -th joint. A joint in the chain is modeled by a location  $\mathbf{m}_i$  and a rotation axis  $\omega_i$ . The exponential map of the corresponding twist  $\xi_i = (-\omega_i \times \mathbf{m}_i, \omega_i)$  yields  $\mathbf{G}_i$  by

$$\mathbf{G}_i = \exp(\theta_i \widehat{\xi}_i). \quad (1)$$

Let  $\mathcal{J}_i \subseteq \{1, \dots, n\}$  be the ordered set of parent joint indices of the  $i$ -th bone. The total rigid motion  $\mathbf{G}_i^{TB}$  of the bone is given by concatenating the global transformation matrix  $\mathbf{G}_0 = \exp(\widehat{\xi}_0)$  and the relative rigid motions matrices  $\mathbf{G}_i$  along the chain by

$$\mathbf{G}_i^{TB} = \mathbf{G}_0 \prod_{j \in \mathcal{J}_i} \exp(\theta_j \widehat{\xi}_j). \quad (2)$$

The rotation part of  $\mathbf{G}_i^{TB}$  is referred to as *tracking bone orientation* of the  $i$ -th bone. In the standard configuration of the kinematic chain, *i.e.*, the zero pose, we choose the local frames of each bone to be coincident with the global frame of reference  $F^T$ . Thus,  $\mathbf{G}_i^{TB}$  also determines the orientation of the bone relative to  $F^T$ . A surface mesh of the actor is attached to the kinematic chain by assigning every vertex of the mesh to one of the bones. Let  $\bar{\mathbf{p}}$  be the homogeneous coordinate of a mesh vertex  $\mathbf{p}$  in the zero pose associated to the  $i$ -th bone. For a configuration  $\mathbf{x}$  of the kinematic chain, the vertex is transformed to  $\bar{\mathbf{p}}'$  using  $\bar{\mathbf{p}}' = \mathbf{G}_i^{TB}\bar{\mathbf{p}}$ .

### 3.3 Optimization Procedure

If several cues are available, *e.g.* image silhouettes and sensor orientation  $\mathbf{z} = (\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}})$ , the likelihood is commonly factored in two independent terms:

$$\arg \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}) = p(\mathbf{z}^{\text{im}}|\mathbf{x})p(\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x}) \quad (3)$$

where it is assumed that the measurements  $\mathbf{z}^{\text{im}}$  and  $\mathbf{z}^{\text{sens}}$  are conditionally independent given that the pose  $\mathbf{x}$  is known. The human pose  $\mathbf{x}$  can then be found by minimizing the negative log-likelihood which yields a weighted combination of cost functions for both terms as in [24]. Since in outdoor scenarios the sensors are not perfectly calibrated and the observations are noisy, fine tuning of the weighting parameters would be necessary to achieve good performance. Furthermore, the orientation information is not used to reduce the state space, and thus the optimization cost and ambiguities. Hence, we propose a different probabilistic formulation of the problem:

$$p(\mathbf{x}|\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}) = \frac{p(\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}})} = \frac{p(\mathbf{z}^{\text{im}}|\mathbf{x})p(\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^{\text{im}})p(\mathbf{z}^{\text{sens}})} \quad (4)$$

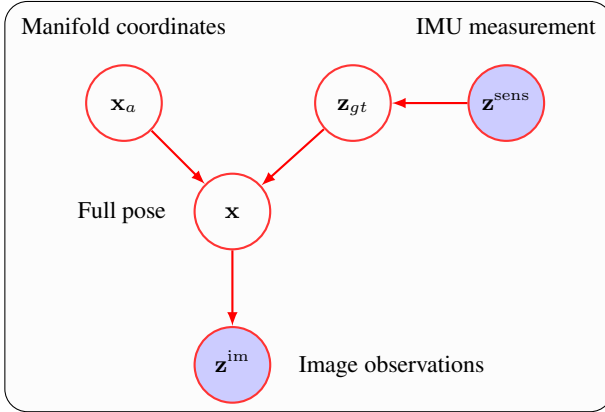
where we assumed independence between sensors and using

$$p(\mathbf{x}|\mathbf{z}^{\text{sens}}) = \frac{p(\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^{\text{sens}})}$$

we obtain the following factorized posterior

$$p(\mathbf{x}|\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}) \propto p(\mathbf{z}^{\text{im}}|\mathbf{x})p(\mathbf{x}|\mathbf{z}^{\text{sens}}). \quad (5)$$

that can be optimized globally and efficiently. We disregard the normalization factor  $p(\mathbf{z}^{\text{im}})$  since it does not depend on the pose  $\mathbf{x}$ . The weighting function  $p(\mathbf{z}^{\text{im}}|\mathbf{x})$  can be modeled by any image-based likelihood function. Our proposed model of  $p(\mathbf{x}|\mathbf{z}^{\text{sens}})$ , as introduced in Sect. 4, integrates uncertainties in the sensor data and constrains the poses to be evaluated to a lower dimensional manifold. For single frame pose estimation, optimization is typically performed by importance sampling, *i.e.* sampling from the prior  $p(\mathbf{x})$  and weighting by the likelihood function  $p(\mathbf{z}^{\text{im}}|\mathbf{x})$ . The problem with this is that the prior is broad compared to  $p(\mathbf{z}^{\text{im}}|\mathbf{x})$  that is peaky and typically multi-valued. By drawing proposals directly from  $p(\mathbf{x}|\mathbf{z}^{\text{sens}})$  we are effectively reducing the number of wasted samples, *i.e.* we are concentrating samples on the likelihood region. For optimization, we use the method proposed in [10]; the implementation details are given in Sect. 4.4.



**Fig. 2.** Graphical model of the approach. The measurements  $\mathbf{z}^{im}$  and  $\mathbf{z}^{sens}$  are shown as shaded nodes because they are observable during inference. The manifold coordinates,  $\mathbf{x}_a$ , the full state pose  $\mathbf{x}$  and the true orientations  $\mathbf{z}_{gt}$  are hidden. To infer the full state pose  $\mathbf{x}$  we optimize the manifold coordinates and marginalize out  $\mathbf{z}_{gt}$ . To integrate out  $\mathbf{z}_{gt}$ , we assume it follows a von-Mises-Fisher distribution with mean direction  $\boldsymbol{\mu} = \mathbf{z}^{sens}$ .

### 4 Manifold Sampling

Assuming that the orientation data  $\mathbf{z}^{sens}$  of the  $N_s$  orientation sensors is accurate and that each sensor has 3 DoF that are not redundant<sup>2</sup>, the  $D$  dimensional pose  $\mathbf{x}$  can be reconstructed from a lower dimensional vector  $\mathbf{x}_a \in \mathbb{R}^d$  where  $d = D - 3N_s$ . In our experiments, a 31 DoF model can be represented by a 16 dimensional manifold using 5 inertial sensors as shown in Fig. 5(a). The mapping is denoted by  $\mathbf{x} = g^{-1}(\mathbf{x}_a, \mathbf{z}^{sens})$  and is described in Sect. 4.1. In this setting, Eq. (3) can be rewritten as

$$\arg \max_{\mathbf{x}_a} p(\mathbf{z}^{im} | g^{-1}(\mathbf{x}_a, \mathbf{z}^{sens})). \tag{6}$$

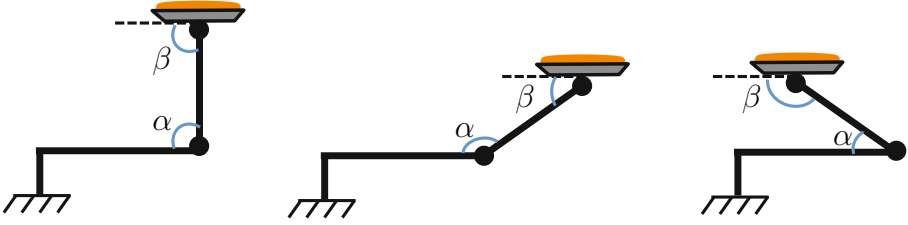
Since the orientation data  $\mathbf{z}^{sens}$  is not always accurate due to sensor noise and calibration errors, we introduce a term  $p(\mathbf{z}_{gt}^{sens} | \mathbf{z}^{sens})$  that models the sensor uncertainty, *i.e.*, the probability of the true orientation  $\mathbf{z}_{gt}^{sens}$  given the sensor data  $\mathbf{z}^{sens}$ . We assume the conditional probability  $p(\mathbf{z}_{gt}^{sens} | \mathbf{z}^{sens})$  follows a *von-Mises Fisher* distribution and it is described in detail Sect. 4.3. Hence, we get the final objective function:

$$\arg \max_{\mathbf{x}_a} \int p(\mathbf{z}^{im} | g^{-1}(\mathbf{x}_a, \mathbf{z}_{gt}^{sens})) p(\mathbf{z}_{gt}^{sens} | \mathbf{z}^{sens}) d\mathbf{z}_{gt}^{sens}. \tag{7}$$

where we marginalize out the sensor noise and optimize the manifold coordinates. The integral can be approximated by importance sampling, *i.e.*, drawing particles from  $p(\mathbf{z}_{gt}^{sens} | \mathbf{z}^{sens})$  and weighting them by  $p(\mathbf{z}^{im} | \mathbf{x})$ . Consequently, we can efficiently concentrate the search space in the neighborhood region of a low dimensional manifold. In addition, we can guarantee that the kinematic constraints are satisfied.

<sup>2</sup> Since the sensors are placed in different body parts they are not redundant because they explain different DoF in the kinematic chain.





**Fig. 3.** Toy example to illustrate our idea to sample from lower dimensional manifolds. For this simple kinematic chain the state vector has 2 *DoF*,  $\mathbf{x} = (\alpha, \beta)$ . If we impose the constraint that the cake plate must be perpendicular to the ground the true state vector has dimensionality 1. The constraint is  $\alpha + \beta = \pi$  and therefore the state vector can be re-parameterized as  $\mathbf{x} = (\alpha, \pi - \alpha)$ . For the problem of human pose estimation however the constraints are non-linear and therefore re-parametrization is achieved by solving small Inverse Kinematics subproblems.

#### 4.1 Inverse Kinematics Using Inertial Sensors

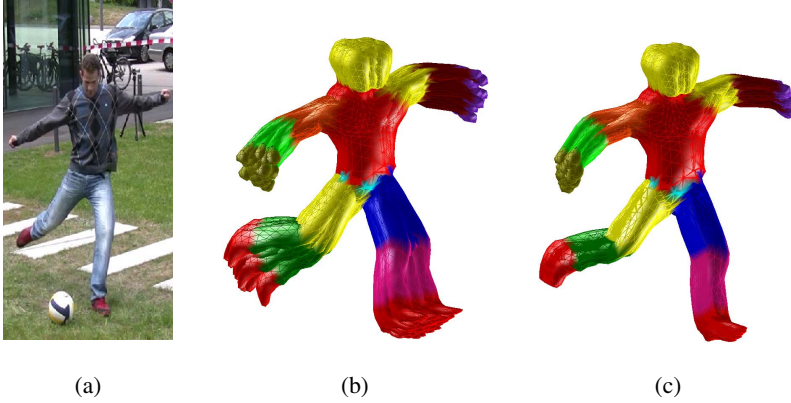
For solving Eq. (7), we derive an analytical solution for the map  $g : \mathbb{R}^D \mapsto \mathbb{R}^{D-3N_s}$  and its inverse  $g^{-1}$ . Here,  $g$  projects  $\mathbf{x} \in \mathbb{R}^D$  to a lower dimensional space and its inverse function  $g^{-1}$  uses the sensor orientations and the coordinates in the lower dimensional space  $\mathbf{x}_a \in \mathbb{R}^{D-3N_s}$  to reconstruct the parameters of the full pose, *i.e.*,

$$g(\mathbf{x}) = \mathbf{x}_a \quad g^{-1}(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) = \mathbf{x}. \quad (8)$$

To derive a set of minimal coordinates, we observe that given the full set of parameters  $\mathbf{x}$  and the kinematic constraints placed by the sensor orientations, a subset of these parameters can be written as a function  $f(\cdot)$  of the others, see Fig. 3 for an intuitive illustration. Specifically, the full set of parameters is decomposed into a set of *active parameters*  $\mathbf{x}_a$  which we want to optimize according to Eq. (7) and a set of *passive parameters*  $\mathbf{x}_p$  that can be derived from the constraint equations and the active set. Writing the state as  $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_p)$  with  $\mathbf{x}_a \in \mathbb{R}^d$  and  $\mathbf{x}_p \in \mathbb{R}^{D-d}$  we have

$$f(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) = \mathbf{x}_p \quad \implies \quad g^{-1}(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) = (\mathbf{x}_a, f(\mathbf{x}_a, \mathbf{z}^{\text{sens}})). \quad (9)$$

Thereby, the direct mapping  $g$  is trivial since from the full set only the active parameters are retained. The inverse mapping  $g^{-1}$  can be found by solving *inverse kinematics* (IK) sub-problems. Several choices for the decomposition into active and passive set are possible. To guarantee the existence of solution for all cases, we choose the passive parameters to be the set of 3 DoF joints that lie on the kinematic branches where a sensor is placed. In our experiments using 5 sensors, we choose the passive parameters to be the two shoulder joints, the two hips and the root joint adding up to a total of 15 parameters which corresponds to  $3N_s$  constraint equations, see Fig. 5(a). Hence, the passive parameters consist of  $N_s$  triplets of joint angles  $\mathbf{x}_p = (\theta_{j_1}, \theta_{j_2}, \theta_{j_3})^T$ ,  $j \in \{1 \dots N_s\}$  with corresponding rotation matrices  $\mathbf{R}_j$ . Since each sensor  $s \in \{1 \dots N_s\}$  is rigidly attached to a bone, there exists a constant rotational offset  $\mathbf{R}_s^{SB}$  between the  $i$ -th bone and the local coordinate system  $F_s^S$  of the sensor attached to it. This offset can



**Fig. 4.** Manifold Sampling: **(a)** Original image. **(b)** Full space sampling. **(c)** Manifold sampling. Note that the generated samples in **(c)** have parallel end-effector orientations because they satisfy the constraints and uncertainty is therefore reduced.

be computed from the tracking bone orientation  $\mathbf{R}_{i,0}^{TB}$  in the first frame and the sensor orientation  $\mathbf{R}_{s,0}^{TS}$

$$\mathbf{R}_s^{SB} = (\mathbf{R}_{s,0}^{TS})^T \mathbf{R}_{i,0}^{TB}. \quad (10)$$

At each frame  $t$ , we obtain *sensor bone orientations*  $\mathbf{R}_{s,t}^{TS} \mathbf{R}_s^{SB}$  by applying the rotational offset. In the absence of sensor noise, it is desired to enforce that the tracking bone orientation and the sensor bone orientation are equal:

$$\mathbf{R}_{i,t}^{TB} = \mathbf{R}_{s,t}^{TS} \mathbf{R}_s^{SB} \quad (11)$$

In Sect. 4.3 we show how to deal with noise in the measurements. Let  $\mathbf{R}_j$  be the relative rotation of the  $j$ -th joint given by the rotational part of Eq. (II). The relative rotation  $\mathbf{R}_j$  associated with the passive parameters can be isolated from Eq. (III). To this end, we expand the tracking bone orientation  $\mathbf{R}_{i,t}^{TB}$  to the product of 3 relative rotations<sup>3</sup>  $\mathbf{R}_j^p$ , the total rotation motion of parent joints in the chain,  $\mathbf{R}_j$ , the unknown rotation of the joint associated with the passive parameters, and  $\mathbf{R}_j^c$ , the relative motion between the  $j$ -th joint and the  $i$ -th joint where the sensor is placed:

$$\mathbf{R}_j^p \mathbf{R}_j \mathbf{R}_j^c = \mathbf{R}_s^{TS} \mathbf{R}_s^{SB} \quad (12)$$

Note that  $\mathbf{R}_j^p$  and  $\mathbf{R}_j^c$  are constructed from the active set of parameters  $\mathbf{x}_a$  using the product of exponentials formula (2). From Eq. (12), we obtain the relative rotation matrix

$$\mathbf{R}_j = (\mathbf{R}_j^p)^T \mathbf{R}_s^{TS} \mathbf{R}_s^{SB} (\mathbf{R}_j^c)^T. \quad (13)$$

Having  $\mathbf{R}_j$  and the known fixed rotation axes  $\omega_{j_1}, \omega_{j_2}, \omega_{j_3}$  of the  $j$ -th joint, the rotation angles  $\theta_{j_1}, \theta_{j_2}, \theta_{j_3}$ , *i.e.*, the passive parameters, must be determined such that

$$\exp(\theta_{j_1} \hat{\omega}_{j_1}) \exp(\theta_{j_2} \hat{\omega}_{j_2}) \exp(\theta_{j_3} \hat{\omega}_{j_3}) = \mathbf{R}_j. \quad (14)$$

<sup>3</sup> The temporal index  $t$  is omitted for the sake of clarity.

This problem can be solved by decomposing it into sub-problems [23], see Sec. 4.2. By solving these sub-problems for every sensor, we are able to reconstruct the full state  $\mathbf{x}$  using only a subset of the parameters  $\mathbf{x}_a$  and the sensor measurements  $\mathbf{z}^{\text{sens}}$ . In this way, the inverse mapping  $g^{-1}(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) = \mathbf{x}$  is fully defined and we can efficiently sample from the manifold, see Fig. 4.

## 4.2 Paden-Kahan Subproblems

We are interested in solving the following problem:

$$\exp(\theta_1 \hat{\omega}_1) \exp(\theta_2 \hat{\omega}_2) \exp(\theta_3 \hat{\omega}_3) = \mathbf{R}_j. \quad (15)$$

This problem can be solved by decomposing it into sub-problems as proposed in [23]. A comprehensive description of the Paden-Kahan subproblems applied to several inverse kinematic problems can also be found in [22]. The basic technique for simplification is to apply the kinematic equations to specific points. By using the property that the rotation of a point on the rotation axis is the point itself, we can pick a point  $\mathbf{p}$  on the third axis  $\omega_3$  and apply it to both sides of Eq. (15) to obtain

$$\exp(\theta_1 \hat{\omega}_1) \exp(\theta_2 \hat{\omega}_2) \mathbf{p} = \mathbf{R}_j \mathbf{p} = \mathbf{q} \quad (16)$$

which is known as the *Paden-Kahan sub-problem 2*. For our problem the 3 rotation axes intersect at the same joint location. Consequently, since we are only interested in the orientations, we can translate the joint location to the origin  $\mathbf{q}_j = O = (0, 0, 0)^T$ . In this way, any point  $\mathbf{p} = \lambda \omega_3$  with  $\lambda \in \mathbb{R}$ ,  $\lambda \neq 0$  is a valid choice for  $\mathbf{p}$ . Eq. (16) can be decomposed in two subproblems

$$\exp(\theta_2 \hat{\omega}_2) \mathbf{p} = \mathbf{c} \quad \text{and} \quad \exp(-\theta_1 \hat{\omega}_1) \mathbf{q} = \mathbf{c}, \quad (17)$$

where  $\mathbf{c}$  is the intersection point between the circles created by the rotating point  $\mathbf{p}$  around axis  $\omega_2$  and the point  $\mathbf{q}$  rotating around axis  $\omega_1$  as shown in Fig. 5(b). In order for Eq. (17) to have a solution, the points  $\mathbf{p}$ ,  $\mathbf{c}$  must lie in the same plane perpendicular to  $\omega_2$ , and  $\mathbf{q}$ ,  $\mathbf{c}$  must lie in the same plane perpendicular to  $\omega_1$ . This implies that the projection of the position vectors  $\mathbf{p}$ ,  $\mathbf{c}$ ,  $\mathbf{q}$  onto the span of  $\omega_1, \omega_2$  respectively must be equal, see Fig. 6

$$\omega_2^T \mathbf{p} = \omega_2^T \mathbf{c} \quad \text{and} \quad \omega_1^T \mathbf{q} = \omega_1^T \mathbf{c} \quad (18)$$

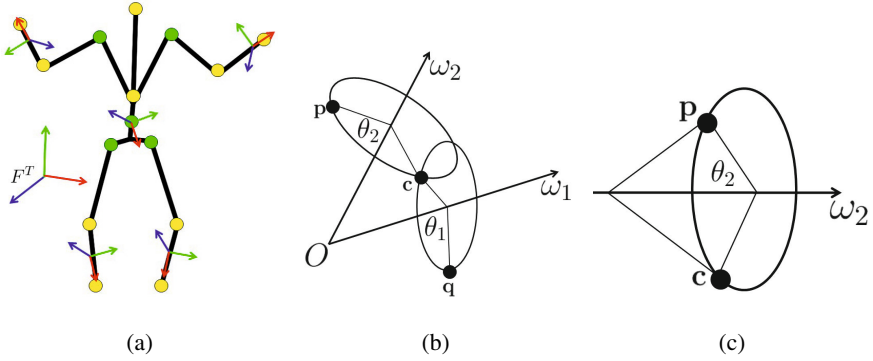
Additionally, the norm of a vector is preserved after rotation and therefore

$$\|\mathbf{p}\| = \|\mathbf{c}\| = \|\mathbf{q}\| \quad (19)$$

Since  $\omega_1$  and  $\omega_2$  are not parallel, the vectors  $\omega_1, \omega_2, \omega_1 \times \omega_2$  form a basis that span  $\mathbb{R}^3$ . Hence, we can write  $\mathbf{c}$  in the new basis as

$$\mathbf{c} = \alpha \omega_1 + \beta \omega_2 + \gamma (\omega_1 \times \omega_2) \quad (20)$$

<sup>4</sup> Since we translated the joint location to the origin we can consider the points as vectors with origin at the joint location  $\mathbf{q}_j$ .



**Fig. 5.** Inverse Kinematics: (a) decomposition into active (yellow) and passive (green) parameters. Paden-Kahan sub-problem 2 (b) and sub-problem 1 (c).

where  $\alpha, \beta, \gamma$  are the new coordinates of  $\mathbf{c}$ . Now, using the fact that  $\omega_2 \perp \omega_1 \times \omega_2$  and  $\omega_1 \perp \omega_1 \times \omega_2$ , we can substitute Eq. (20) into Eq. (18) to obtain a system of two equations with two unknowns ( $\alpha, \beta$ )

$$\begin{aligned}\omega_2^T \mathbf{p} &= \alpha \omega_2^T \omega_1 + \beta \\ \omega_1^T \mathbf{q} &= \alpha + \beta \omega_1^T \omega_2\end{aligned}\quad (21)$$

from which we can isolate the first two coordinates of  $\mathbf{c}$

$$\begin{aligned}\alpha &= \frac{(\omega_1^T \omega_2) \omega_2^T \mathbf{p} - \omega_1^T \mathbf{q}}{(\omega_1^T \omega_2)^2 - 1} \\ \beta &= \frac{(\omega_1^T \omega_2) \omega_1^T \mathbf{q} - \omega_2^T \mathbf{p}}{(\omega_1^T \omega_2)^2 - 1}\end{aligned}\quad (22)$$

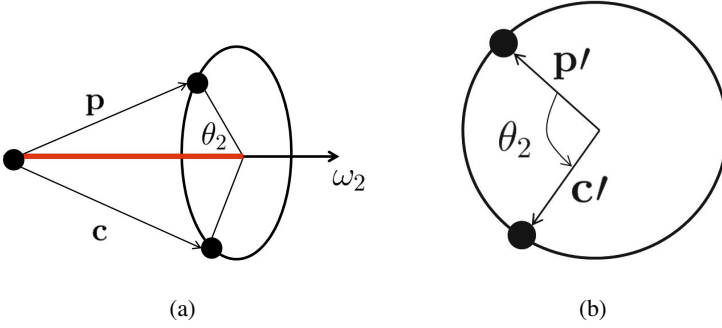
From Eq. (19) and Eq. (20) we can write

$$\|\mathbf{p}\|^2 = \|\mathbf{c}\|^2 = \alpha^2 + \beta^2 + 2\alpha\beta\omega_1^T \omega_2 + \gamma^2 \|\omega_1 \times \omega_2\|^2 \quad (23)$$

and obtain the third coordinate  $\gamma$  as

$$\gamma^2 = \frac{\|\mathbf{p}\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta\omega_1^T \omega_2}{\|\omega_1 \times \omega_2\|^2} \quad (24)$$

This last equation has no solution when the circles do not intersect, one solution when the circles are tangential and two solutions when the circles intersect at two points. For our choice of decomposition, the passive parameters correspond to  $3DoF$  joints which are modeled as 3 concatenated revolute joints whose axis are mutually orthogonal. Therefore, there always exists a solution [22]. We note that the inverse kinematic solutions presented here are also valid for other decompositions, *e.g.* one could choose as passive parameters two rotation axes of the shoulder joint and one rotation axis of



**Fig. 6.** Paden-Kahan subproblem 1: (a) the projection length of  $\mathbf{p}$  and  $\mathbf{c}$  onto  $\omega_2$  must be equal, (b) the projection of the vectors  $\mathbf{p}$  and  $\mathbf{c}$  onto the orthogonal plane to the rotation axes  $\omega_2$

the elbow joints. However, the existence of solution should then be checked during the sampling process. Once we have the new coordinates  $(\alpha, \beta, \gamma)$  we can obtain the intersection point  $\mathbf{c}$  in the original coordinates using equation Eq. (20). Thereafter, Eq. (17) can be decomposed into two problems of the form

$$\begin{aligned} \exp(\theta_2 \hat{\omega}_2) \mathbf{p} &= \mathbf{c} \\ \exp(-\theta_1 \hat{\omega}_1) \mathbf{q} &= \mathbf{c} \end{aligned} \quad (25)$$

which simplifies to finding the rotation angle about a fixed axis that brings a point  $\mathbf{p}$  to a second one  $\mathbf{c}$ , which is known as *Paden-Kahan sub-problem 1*

$$\exp(\theta_2 \hat{\omega}_2) \mathbf{p} = \mathbf{c}. \quad (26)$$

This problem has a solution when the projections of the vectors  $\mathbf{p}$  and  $\mathbf{c}$  onto the orthogonal plane to  $\omega_2$  have equal lengths. Let  $\mathbf{p}'$  and  $\mathbf{c}'$  be the projections of  $\mathbf{p}$ ,  $\mathbf{c}$  onto the plane perpendicular to  $\omega_2$ , see Fig. 6.

$$\mathbf{p}' = \mathbf{p} - \omega_2 \omega_2^T \mathbf{p} \quad \text{and} \quad \mathbf{c}' = \mathbf{c} - \omega_2 \omega_2^T \mathbf{c}. \quad (27)$$

If the projections have equal lengths  $\|\mathbf{p}'\| = \|\mathbf{c}'\|$  then the problem is as simple as finding the angle between the two vectors

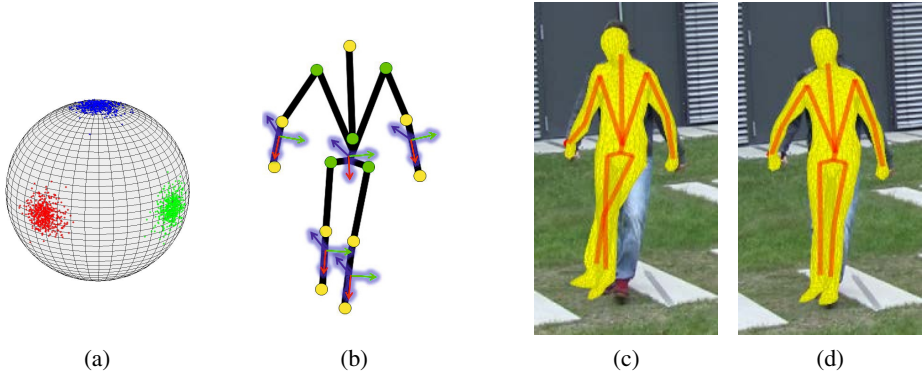
$$\begin{aligned} \omega_2^T (\mathbf{p}' \times \mathbf{c}') &= \sin \theta_2 \|\mathbf{p}'\| \|\mathbf{c}'\| \\ \mathbf{p}' \cdot \mathbf{c}' &= \cos \theta_2 \|\mathbf{p}'\| \|\mathbf{c}'\| \end{aligned} \quad (28)$$

By dividing the equations we finally obtain the rotation angle using the arc tangent

$$\theta_2 = \text{atan2}(\omega_2^T (\mathbf{p}' \times \mathbf{c}'), \mathbf{p}' \cdot \mathbf{c}'). \quad (29)$$

We can find  $\theta_1$  using the same procedure. Finally,  $\theta_3$  is obtained from Eq. (15) after substituting  $\theta_1$  and  $\theta_2$

$$\exp(\theta_3 \hat{\omega}_3) = \exp(\theta_1 \hat{\omega}_1)^T \exp(\theta_2 \hat{\omega}_2)^T \mathbf{R}_j = \mathbf{R} \quad (30)$$



**Fig. 7.** Sensor noise model. **(a)** Points disturbed with rotations sampled from a von Mises-Fisher distribution. **(b)** The orientation of the particles can deviate from the sensor measurements. Tracking without **(c)** and with **(d)** sensor noise model.

where the rotation matrix  $\mathbf{R}$  is known. The rotation angle  $\theta_3$  satisfies

$$2 \cos \theta_3 = (\text{trace}(\mathbf{R}) - 1) \tag{31}$$

$$2 \sin \theta_3 = \omega_3^T \mathbf{r} \tag{32}$$

where  $\mathbf{r} = (\mathbf{R}_{32} - \mathbf{R}_{23}, \mathbf{R}_{13} - \mathbf{R}_{31}, \mathbf{R}_{21} - \mathbf{R}_{12})$  (page 584 of [14]). Finally, the rotation angle  $\theta_3$  can be computed from  $\cos \theta_3$  and  $\sin \theta_3$  using  $\text{atan2}$ . By solving these sub-problems for every sensor, we are able to reconstruct the full state  $\mathbf{x}$  using only a subset of the parameters  $\mathbf{x}_a$  and the sensor measurements  $\mathbf{z}^{\text{sens}}$ . The good property of this geometric algorithms for solving inverse kinematics is that they are numerically very stable. More importantly, the same principle can be applied to solve more complex IK problems involving a number of positional and orientational constraints.

### 4.3 Sensor Noise Model

In practice, perfect alignment and synchronization of inertial and video data is not possible. In fact, there are at least four sources of uncertainty in the inertial sensor measurements, namely inherent sensor noise from the device, temporal unsynchronization with the images, small alignment errors between the tracking coordinate frame  $F^T$  and the inertial frame  $F^I$ , and errors in the estimation of  $\mathbf{R}_s^{SB}$ . Hence, we introduce a noise model  $p(\mathbf{z}_{gt}^{\text{sens}} | \mathbf{z}^{\text{sens}})$  in our objective function (7). Rotation errors are typically modeled by assuming that the measured rotations are distributed according to a Gaussian in the tangent spaces which is implemented by adding Gaussian noise  $v^i$  on the parameter components, *i.e.*,  $\tilde{\mathbf{x}}_j = \mathbf{x}_j + v^i$ . The topological structure of the elements, a 3-sphere  $S^3$  in case of quaternions, is therefore ignored. The *von Mises-Fisher* (MF) distribution models errors of elements that lie on a unit sphere  $S^{p-1}$  [8] and is defined as

$$f_p(\mathbf{x}; \boldsymbol{\mu}, \kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{d/2-1}(\kappa)} \exp(\kappa \boldsymbol{\mu}^T \mathbf{x}) \tag{33}$$

where  $I_\nu$  denotes the modified Bessel function of the first kind,  $\boldsymbol{\mu}$  is the mean direction, and  $\kappa$  is a concentration parameter that determines the dispersion from the true position. The distribution is illustrated in Fig. 7. For our problem,  $p = 4$  and thus the elements  $\mathbf{x}$  are quaternions. Therefore, on the one hand samples of the MF distribution are quaternions whose corresponding axis of rotation are uniformly distributed in all directions. On the other hand, the sample concentration decays with the angle of rotation. To see this, observe that the distribution can be expressed as a function of the angular rotation  $\theta$  from the mean  $\boldsymbol{\mu}$  where we replaced the inner product  $\boldsymbol{\mu}^T \mathbf{x}$  by  $\cos\left(\frac{\theta}{2}\right)$  (the inner product between two quaternions results in  $\cos\left(\frac{\theta}{2}\right)$ , where  $\theta$  is the geodesic angle distance between rotations).

In order to approximate the integral in Eq. (7) by importance sampling, we use the method proposed in [39] to draw samples  $\mathbf{q}_w$  from the von Mises-Fisher distribution with  $p = 4$  and  $\boldsymbol{\mu} = (1, 0, 0, 0)^T$ , which is the quaternion representation of the identity. We use a fixed dispersion parameter of  $\kappa = 1000$ . The sensor quaternions are then rotated by the random samples  $\mathbf{q}_w$ :

$$\tilde{\mathbf{q}}_s^{TS} = \mathbf{q}_s^{TS} \circ \mathbf{q}_w \quad (34)$$

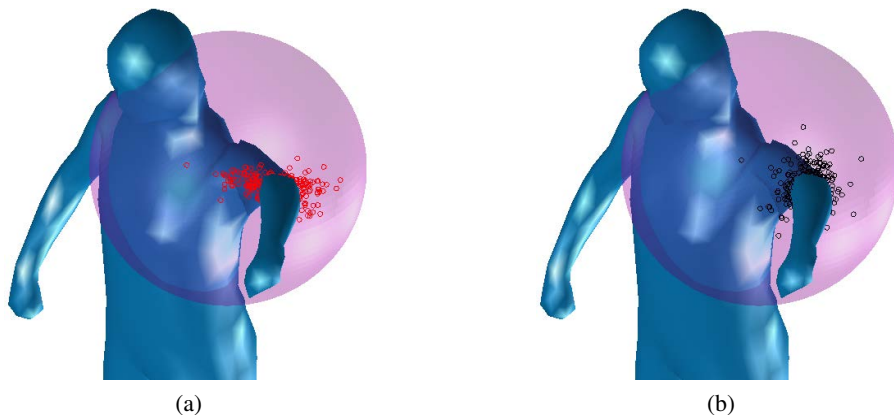
where  $\circ$  denotes quaternion multiplication. In this way, for every particle, samples  $\tilde{\mathbf{q}}_s^{TS}$  are drawn from  $p(\mathbf{z}_{gt}^{\text{sens}} | \mathbf{z}^{\text{sens}})$  using Eq. (34) obtaining a set of distributed measurements  $\tilde{\mathbf{z}}^{\text{sens}} = (\tilde{\mathbf{q}}_1^{TS} \dots \tilde{\mathbf{q}}_{N_s}^{TS})$ . This can be interpreted as the analogous of additive Gaussian Noise where  $\mathbf{q}_w$  is a rotation noise sample. Thereafter, the full pose is reconstructed from the newly computed orientations with  $g^{-1}(\mathbf{x}_a, \tilde{\mathbf{z}}^{\text{sens}})$  as explained in Sect. 4.1 and weighted by  $p(\mathbf{z}^{\text{im}} | \mathbf{x})$ .

In Fig. 8, we compare the inverse kinematic solutions of 500 samples  $i \in \{1 \dots 500\}$  by simply adding Gaussian noise *only* on the passive parameters  $\{g^{-1}(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) + \mathbf{v}^i\}_i$  and by modeling sensor noise with the von Mises-Fisher distribution  $\{g^{-1}(\mathbf{x}_a, \tilde{\mathbf{z}}^{\text{sens}, i})\}_i$ . For the generated samples, we fixed the vector of manifold coordinates  $\mathbf{x}_a$  and we used equivalent dispersion parameters for both methods. To visualize the reconstructed poses we only show, for each sample, the elbow location represented as a point in the sphere. This example shows that simply adding Gaussian noise on the parameters is biased towards one direction that depends on the current pose  $\mathbf{x}$ . By contrast, the samples using von Mises-Fisher are uniformly distributed in all directions and the concentration decays with the angular error from the mean. Note, however, that Fig. 8 is a 3D visualization, in reality the bone orientations of the reconstructed poses should be visualized as points in a 3-sphere  $S^3$ .

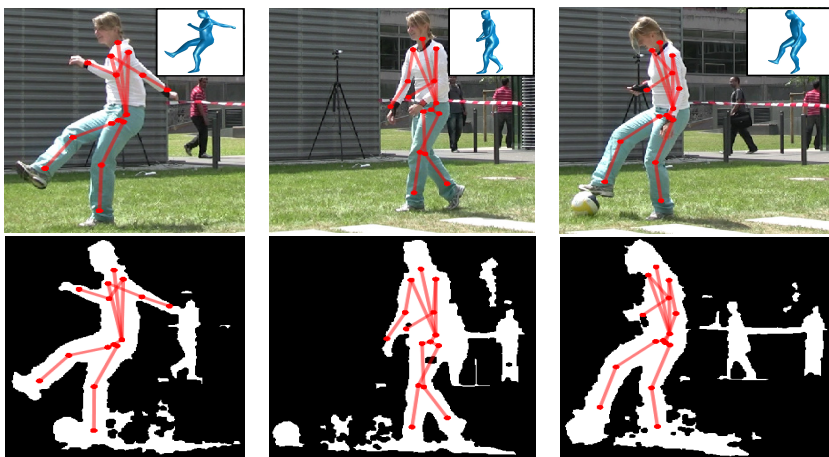
$$f_p(\theta; \kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{d/2-1}(\kappa)} \exp\left(\kappa \cos\left(\frac{\theta}{2}\right)\right) \quad (35)$$

#### 4.4 Implementation Details

To optimize Eq. (7), we have implemented ISA (Interacted Simulated Annealing), the global optimization approach that has been proposed in [10] and use only the first stage of



**Fig. 8.** Sensor noise model. 500 samples of the IK elbow location are shown as points using: (a) added Gaussian noise and (b) noise from the von Mises-Fisher distribution.



**Fig. 9.** Tracking with background clutter

the algorithm, *i.e.* we do not locally optimize. ISA is based on simulated annealing which is a stochastic optimization technique to locate a good approximation of the global optimum of a cost function in a large search space. In the remainder of this paper we will use the term global optimization whenever ISA was used for optimization to make the distinction with local optimization methods. As cost function, we use the silhouette and color terms

$$V(\mathbf{x}) = \lambda_1 V_{silh}(\mathbf{x}) + \lambda_2 V_{app}(\mathbf{x}) \quad (36)$$



with the setting  $\lambda_1 = 2$  and  $\lambda_2 = 40$ . Although a good likelihood model is essential for good performance, it is not the focus of our work and we refer the interested reader to [26] for more details. During tracking, the initial particles  $\{\mathbf{x}_a^i\}_i$  are predicted from the particles in the previous frame using a 3rd order autoregression and projected to the low-dimensional manifold using the mapping  $g$ ; see Sect. 4.1. The optimization is performed only over the active parameters  $\mathbf{x}_a \in \mathbb{R}^{D-3N_s}$ , *i.e.*, the diffusion step is performed in  $\mathbb{R}^{D-3N_s}$ . Specifically, diffusion is performed with a Gaussian kernel with zero mean and covariance matrix

$$\Sigma_{a,k} = \frac{\alpha_\Sigma}{N-1} \left( \rho \mathbf{I} + \sum_i^N (\mathbf{x}_{a,k}^{(i)} - \mu_{a,k})(\mathbf{x}_{a,k}^{(i)} - \mu_{a,k})^T \right) \quad (37)$$

proportional to the sampling covariance matrix scaled by  $\alpha_\Sigma$  where  $\mu_k$  is the particle set mean at the current iteration  $k$ .

For the weighting step, we use the approach described in Sect. 4.3 to generate a sample  $\tilde{\mathbf{z}}^{\text{sens},i}$  from  $p(\mathbf{z}_{gt}^{\text{sens}} | \mathbf{z}^{\text{sens}})$  for each particle  $\mathbf{x}_a^i$ . Consequently, we can map each particle back to the full space using  $\mathbf{x}^i = g^{-1}(\mathbf{x}_a^i, \tilde{\mathbf{z}}^{\text{sens},i})$  and weight it by

$$\pi_k^{(i)} = \exp(-\beta_k \cdot V(g^{-1}(\mathbf{x}_{a,k}^i, \tilde{\mathbf{z}}^{\text{sens},i}))), \quad (38)$$

where  $\beta_k$  is the inverse temperature of the annealing scheme at iteration  $k$  and  $V(\cdot)$  is the image cost function defined in Eq. (36). From the obtained set of weighted particles  $\{\pi_k^{(i)}, \mathbf{x}_{a,k}^{(i)}\}_{i=1}^N$  we draw a new set of particles with resampling and probability equal to the normalized weights. The weighting, resampling and diffusion step are iterated  $M$  times before going to the next frame. In our experiments, we used 15 iterations for optimization. Finally, the pose estimate is obtained from the remaining particle set at the last iteration as

$$\hat{\mathbf{x}}_t = \sum_i \pi_k^{(i)} g^{-1}(\mathbf{x}_{a,k}^{(i)}, \tilde{\mathbf{z}}^{\text{sens},i}). \quad (39)$$

The steps of our proposed sampling scheme are outlined in Algorithm 1.

**Dynamics:** To model the dynamics we use a 3rd order auto-regression using Gaussian Process regression that provides a prediction  $\mathbf{x}^{\text{pred}}$  and a covariance matrix  $\Sigma^{\text{pred}}$  related with the confidence of the prediction. Thereby, the particles from the previous frame are drifted towards the predicted mean  $\mathbf{x}^{\text{pred}}$  and diffused with a Gaussian kernel with zero mean and covariance  $\Sigma^{\text{pred}}$ . In order to obtain the low dimensional particle set, every particle is projected  $g(\mathbf{x}_t^i) = \mathbf{x}_{a,t}^{(i)}$ <sup>5</sup>. We note that we do not learn a mapping directly in the low dimensional space since the previous estimates of passive parameters  $\mathbf{x}_{p,t-4:t-1}$  are in general also correlated with the active parameters  $\mathbf{x}_{a,t}$ . The particle set is used as the initial proposal distribution for the first iteration of ISA.

<sup>5</sup> Since the basic Gaussian process does not take the correlation of the output variables into account the process is equivalent to a 3rd order regression from previous full state estimates to the manifold coordinates.

**Algorithm 1.** Proposed algorithm

**Require:** number of layers  $M$ , number of samples  $N$ , initial distribution  $\mathcal{L}_0$ , sensor orientations  $\mathbf{z}^{\text{sens}}$ , image cost function  $V(\cdot)$

Initialize: Draw  $N$  initial samples from  $\mathcal{L}_0 \rightarrow \mathbf{x}_{a,k}^{(i)}$

**for** layer  $k = 0$  to  $M$  **do**

1. *MANIFOLD SAMPLING*

start from the set of un-weighted particles of the previous layer

**for**  $i = 1$  to  $N$  **do**

1.1 *SENSOR NOISE*

*/\* draw a sample  $\tilde{\mathbf{z}}^{\text{sens},i}$  from  $p(\mathbf{z}_{gt}^{\text{sens}} | \mathbf{z}^{\text{sens}})$  \*/*

**for**  $s = 1$  to  $N_s$  **do**

draw sample from von-Mises Fisher  $f_p(\boldsymbol{\mu}, \kappa) \rightarrow \mathbf{q}_w$

$\tilde{\mathbf{q}}_s^{TS} = \mathbf{q}_s^{TS} \circ \mathbf{q}_w$

**end for**

set  $\tilde{\mathbf{z}}^{\text{sens},i} = (\tilde{\mathbf{q}}_1^{TS} \dots \tilde{\mathbf{q}}_{N_s}^{TS})^T$

1.1 *INVERSE KINEMATICS*

*/\* computation of  $\mathbf{x}_k^{(i)} = g^{-1}(\mathbf{x}_{a,k}^i, \tilde{\mathbf{z}}^{\text{sens}})$  \*/*

**for**  $j = 1$  to  $N_s$  **do**

compute:  $\mathbf{R}_s^{TS} = \text{quat2mat}(\tilde{\mathbf{q}}_j^{TS})$

compute:  $\mathcal{F}(\mathbf{x}_a) \rightarrow \mathbf{R}_j^p, \mathbf{R}_j^c$

set:  $\mathbf{R}_j = (\mathbf{R}_j^p)^T \mathbf{R}_s^{TS} \mathbf{R}_s^{SB} (\mathbf{R}_j^c)^T$

solve:  $\exp(\theta_{j1} \hat{\omega}_{j1}) \exp(\theta_{j2} \hat{\omega}_{j2}) \exp(\theta_{j3} \hat{\omega}_{j3}) = \mathbf{R}_j$

**end for**

set:  $\pi_k^{(i)} = \exp\left(-\beta_k \cdot V\left(\mathbf{x}_k^{(i)}\right)\right)$

**end for**

set:  $\mathcal{L}_k = \{\pi_k^{(i)}, \mathbf{x}_{a,k}^{(i)}\}_{i=1}^N$

2. *RESAMPLING*

draw  $N$  samples from  $\mathcal{L}_k \rightarrow \mathbf{x}_{a,k}^{(i)}$

3. *DIFFUSION*

$\mathbf{x}_{a,k+1}^{(i)} = \mathbf{x}_{a,k}^{(i)} + \mathbf{B}_k \quad \{\mathbf{B}_k \text{ is a sample from } \mathcal{N}(0, \Sigma_a)\}$

**end for**

## 5 Experiments

The standard benchmark for human motion capture is *HumanEva* that consists of indoor sequences. However, no outdoor benchmark data comprising video as well as inertial data exists for free use yet. Therefore, we recorded eight sequences of two subjects performing four different activities, namely walking, karate, basketball and soccer. Multiview image sequences are recorded using four unsynchronized off-the-shelf video cameras. To record orientation data, we used an Xsens Xbus Kit [36] with 10 sensors. Five of the sensors, placed at the lower limbs and the back, were used for tracking, and five of the sensors, placed at the upper limbs and at the chest, were used for validation. As for any comparison measurements taken from sensors or marker-based systems, the accuracy of the validation data is not perfect but is useful to evaluate the performance of a given approach. The eight sequences in the data set comprise over 3 minutes of footage sampled at 25 Hz. Note that the sequences are significantly more difficult than the

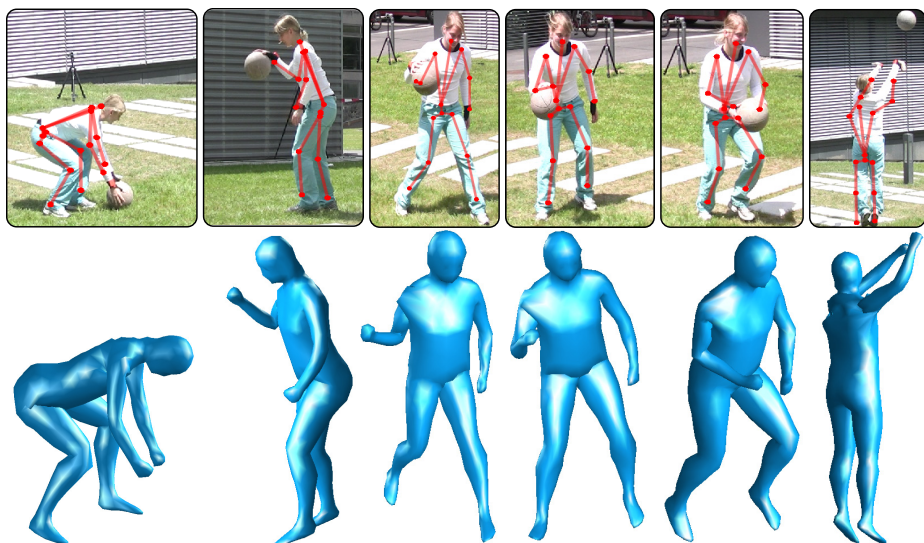


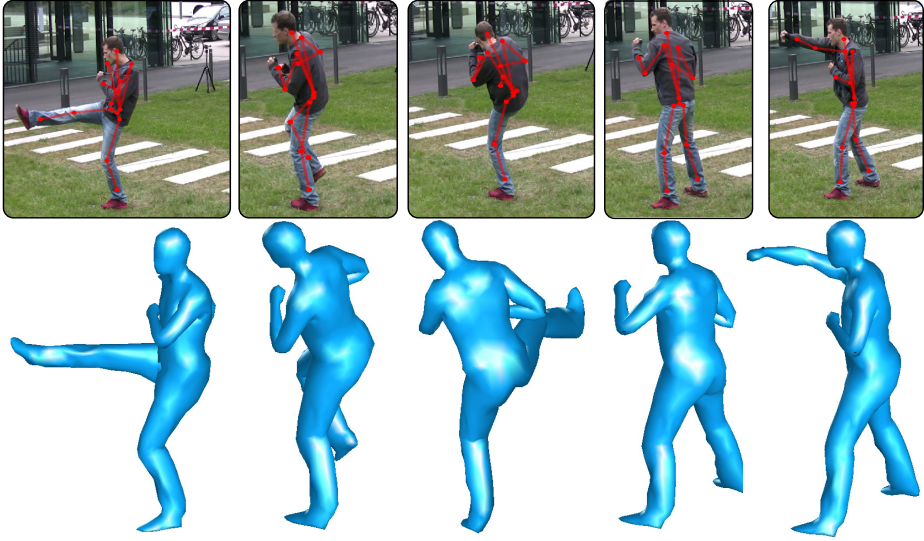
Fig. 10. Tracking with strong illumination

sequences of *HumanEva* since they include fast motions, illumination changes, shadows, reflections and background clutter. For the validation of the proposed method, we additionally implemented five baseline trackers: two video-based trackers based on local (L) and global optimization (G) respectively and three hybrid trackers that also integrate orientation data: local optimization (LS), global optimization (GS) and rejection sampling (RS) which we briefly describe here

- (L): Local optimization tracker. The model is projected to the image to find correspondences between the image silhouette contours and the model points. Then, the non-linear least squares problem is solved using a variant of *Levenberg-Marquardt algorithm*, see [15,25] for more details.
- (G): Global Particle based optimization. Optimization here is performed by means of simulated annealing, *i.e.*, pose hypotheses are generated and weighted with progressively smooth versions of the image likelihood. The final pose is obtained as the average of the particle set in the last annealing layer, see [6,10] for more details.
- (LS): Local optimization + inertial Sensors. Optimization is again performed by means of non-linear least squares but the cost function to be minimized consists of an image term and a term that models the likelihood of the inertial sensor measurements

$$V(\mathbf{x}) = \mu_1 V_1^{\text{im}}(\mathbf{x}) + \mu_2 V_1^{\text{sens}}(\mathbf{x})$$

where  $V_1^{\text{sens}}(\mathbf{x})$  is defined as the squared Frobenious norm between the sensor and the tracking bone orientation matrices. Both the model-image Jacobian and the orientational Jacobian are derived analytically for better accuracy and efficiency. The algorithm is based on [24].



**Fig. 11.** Tracking results of a karate sequence

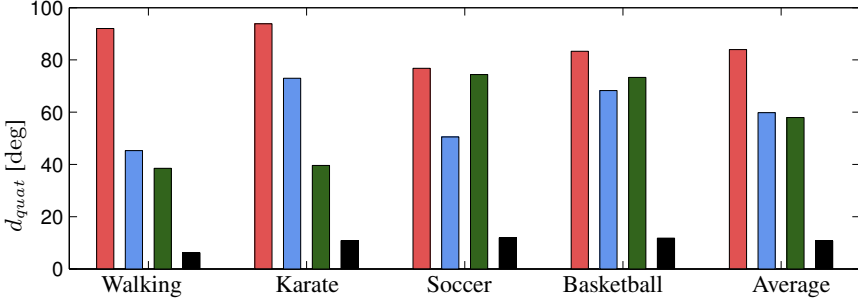
- (GS): Global particle based optimization with Sensors. Like the (G) method but including the inertial sensor measurements in the weighting function. We optimize a cost function

$$V(\mathbf{x}) = \mu_1 V^{\text{im}}(\mathbf{x}) + \mu_2 V_2^{\text{sens}}(\mathbf{x})$$

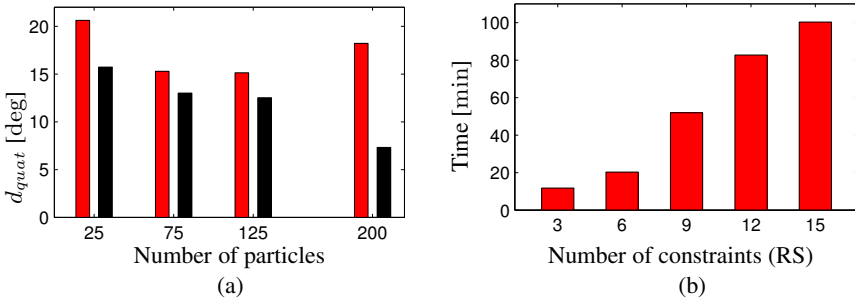
where the image term  $V^{\text{im}}(\mathbf{x})$  is the one defined in Eq. (36) and is chosen to be a piece-wise increasing linear function of the angular error between the tracking and the sensor bone orientations. That is, for angular errors bigger than 10 degrees we scale the cost by a factor of 5. Big deviations from the orientation measurement could in principle be penalized with a quadratic function but this yields to many particles being rejected in early stages and results in lower performance. Note that although  $\mu_2 V_2^{\text{sens}}(\mathbf{x})$  and  $\mu_2 V_1^{\text{sens}}(\mathbf{x})$  are not identical they are both functions of distance metrics for rotations and are thus equivalent. For (LS) we optimize  $\mu_2 V_1^{\text{sens}}(\mathbf{x})$  because derivatives are easier to compute. We hand tuned the influence weights  $\mu_1, \mu_2$  to obtain the best possible performance.

- (RS): Rejection Sampling. This method is commonly used to sample hypotheses that satisfy a set of constraints. The method works by sampling hypotheses and rejecting hypotheses that do not satisfy the constraints up to a certain tolerance. It was for example used in [17] to integrate object interaction constraints. For our problem, to combine inertial data with video images we draw particles directly from  $p(\mathbf{x}_t | \mathbf{z}^{\text{sens}})$  using a rejection sampling scheme. In our implementation of (RS), we reject a particle when the angular error for any of the constraints is bigger than 10 degrees.

For a comprehensive overview of model based methods for human pose estimation we refer the interested reader to [26].



**Fig. 12.** Mean orientation error of our 8 sequences (2 subjects) for methods (bars left to right) L (local optimization), LS (local+sensors), GL (global optimization), and ours P

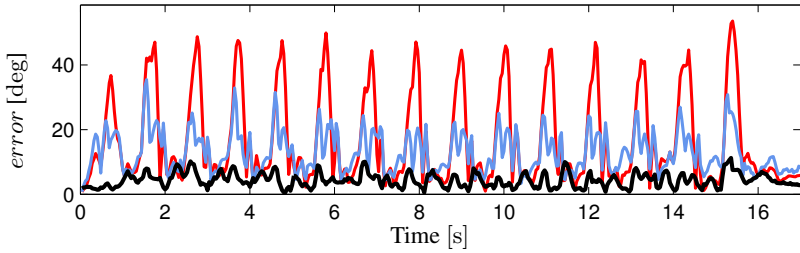


**Fig. 13.** (a): Orientation error with respect to number of particles with (red) the GS method and (black) our algorithm. (b): Running time of *rejection sampling* (RS) with respect to number of constraints. By contrast our proposed method takes 0.016 seconds for 15 *DoF* constraints. The time to evaluate the image likelihood is excluded as it is independent of the algorithm.

Let the *validation set* be the set of quaternions representing the sensor bone orientations *not* used for tracking as  $\mathbf{v}^{\text{sens}} = \{\mathbf{q}_1^{\text{val}}, \dots, \mathbf{q}_5^{\text{val}}\}$ . Let  $i_s, s \in \{1 \dots t\}$  be the corresponding bone index, and  $\mathbf{q}_{i_s}^{TB}$  the quaternions of the tracking bone orientation (Sect. 3.2). We define the *error measure* as the average geodesic angle between the sensor bone orientation and the tracking orientation for a sequence of  $T$  frames as

$$d_{quat} = \frac{1}{5T} \sum_{s=1}^5 \sum_{t=1}^T \frac{180^\circ}{\pi} 2 \arccos |\langle \mathbf{q}_s^{\text{val}}(t), \mathbf{q}_{i_s}^{TB}(t) \rangle|. \quad (40)$$

**Comparison with Video and Local Trackers:** We compare the performance of four different tracking algorithms using the distance measure, namely (L), (G), (LS) and our proposed approach (P). We show  $d_{quat}$  for the eight sequences and each of the four trackers in Fig. 12. For (G) and (P) we used the same number of particles  $N = 200$ . As it is apparent from the results, local optimization is not suitable for outdoor scenes as it gets trapped in local minima almost immediately. Our experiments show that LS as proposed in [24] works well until there is a tracking failure in which case the tracker recovers only by chance. Even using (G), the results are unstable since the video-based cues are too



**Fig. 14.** Angular error for the left hip of a walking motion with (red) no sensor noise model (NN), (blue) Gaussian noise model (GN) and (black) our proposed (MFN)



**Fig. 15.** Tracking results of a soccer sequence

ambiguous and the motions too fast to obtain reliable pose estimates. By contrast, our proposed tracker achieves an average error of  $10.78^\circ \pm 8.5^\circ$  and clearly outperforms the pure video-based trackers and (LS).

**Comparison with GS:** In Fig. 13(a), we show  $d_{quat}$  for a varying number of particles using the (GS) and our proposed algorithm (P) for a walking sequence.

The error values show that optimizing a combined cost function leads to bigger errors for the same number of particles when compared to our method. This was an expected result since we reduce the dimension of the search space by sampling from the manifold and consequently less particles are needed for equal accuracy. Most importantly, the visual quality of the 3D animation deteriorates more rapidly with (GS) as the number of particles are reduced<sup>6</sup>. This is partly due to the fact that the constraints are not always satisfied when additional error terms guide the optimization.

<sup>6</sup> See the video for a comparison of the estimated motions at

<http://www.tnt.uni-hannover.de/~pons/>

**Comparison with Rejection Sampling (RS):** Another option for combining inertial data with video images is to draw particles directly from  $p(\mathbf{x}_t | \mathbf{z}^{\text{sens}})$  using a simple rejection sampling scheme. In our implementation of (RS), we reject a particle when the angular error is bigger than 10 degrees. Unfortunately, this approach can be very inefficient especially if the manifold of poses that fulfill the constraints lies in a narrow region of the parameter space. This is illustrated in Fig. 13(b) where we show the processing time per frame (excluding image likelihood evaluation) using 200 particles as a function of the number of constraints. Unsurprisingly, rejection sampling does not scale well with the number of constraints taking as much as 100 minutes for 15 DoF constraints imposed by the 5 sensors. By contrast, our proposed sampling method takes in the worst case (using 5 sensors) 0.016 seconds per frame. These findings show that sampling directly from the manifold of valid poses is a much more efficient alternative.

**Sensor Noise Model:** To evaluate the influence of the sensor noise model, we tracked one of the walking sequences in our dataset using no noise (NN), additive Gaussian noise (GN) in the passive parameters and noise from the von Mises-Fisher (MFN) distribution as proposed in Sect. 4.3. In Fig. 14 we show the angular error of the left hip using each of the three methods. With (NN) error peaks occur when the left leg is matched with the right leg during walking, see Fig. 7. This typical example shows that slight misalignment (as little as  $5^\circ - 10^\circ$ ) between video and sensor data can miss-guide the tracker if no noise model is used. The error measure was  $26.8^\circ$  with no noise model,  $13^\circ$  using Gaussian noise and  $7.3^\circ$  with the proposed model. The error is reduced by 43% with (MFN) compared to (GN) which indicates that the von Mises-Fisher is a more suited distribution to explore orientation spaces than the commonly used Gaussian. This last result might be of relevance not only to model sensor noise but to any particle-based HMC approach. Finally, pose estimation results for typical sequences of our dataset are shown in Fig. 9, 10, 11 and 15. A video of the proposed approach along with tracking results can be found in the authors website<sup>7</sup>.

## 6 Discussion and Limitations

State-of-the-art video trackers, either based on local or global optimization, suffer from 3D ambiguities inherent in video and usually fail to recover from errors. Our experiments reveal that video based pose estimation algorithms benefit from using a set of small IMUs, specially in outdoor scenarios where the image observation models are weak and ambiguous. Nonetheless, combining inertial and video measurements poses a difficult optimization problem that has to be dealt efficiently. Local optimization is fast and accurate in indoor scenarios. However, our findings indicate that to integrate orientation, (LS) is not suited in outdoor scenarios because it suffers from tracking failures that occur frequently. Optimizing a global cost function (GS) is also not the best choice since it yields an optimization in a high dimensional space which is computationally more expensive. In particular, a high number of hypotheses have to be generated since the search space volume is huge. Rejection sampling (RS) is not suited because it scales

<sup>7</sup><http://www.tnt.uni-hannover.de/~pons/>

very poorly with the number of constraints and the computational time grows exponentially. Finally, we showed that the commonly used Gaussian Noise is outperformed by the proposed von Mises-Fisher noise model when it comes to modeling orientation ambiguities. The reason is that spherical sampling in the joint angle domain does not yield spatially spherical joint configurations as opposed to sampling using (MF). Our proposed method overcomes much of the described limitations: on the one hand the search space is explored only in the region that satisfies the constraints, and on the other hand sampling using Inverse Kinematics has a reinitialization power that overcomes tracking failures in many occasions. Unfortunately, the proposed method is limited by the availability of IMUs. Even though the IMUs are very small and we use only five, they are unavailable in several applications such as surveillance or MoCap and scene understanding from video archives. Another issue that requires improvement is robustness to unsynchronization produced by the IMUs lag during fast motions. The performance of our proposed tracker is still affected from such unsynchronization between IMUs and the video cameras. Since IMUs do not provide any positional measurement, our tracker fails when the body limbs (specially the arms) are not detectable due to long term occlusions. Finally, even though we achieve considerable computational gains w.r.t optimizing the full state space, evaluating the image cost function for every sample is still a bottle neck. To further reduce computational time, an option would be to use very few particles *e.g.* 25 and then locally optimize to obtain better accuracy. Although in this work we have presented an algorithm to combine IMUs with video, the ideas shown here are of significant relevance for the computer vision community. Firstly, the Inverse Kinematics sampling scheme can be used to generate pose hypotheses that satisfy a set of kinematic constraints (we leave extensions to positional constraints as interesting future work). Secondly, the proposed sensor noise model can be used in any problem that involves modeling or optimization of rotation elements.

## 7 Conclusions

By combining video with IMU input, we introduced a novel particle-based hybrid tracker that enables robust 3D pose estimation of arbitrary human motions in outdoor scenarios. As the two main contributions, we first presented an analytic procedure based on Inverse Kinematics for efficiently sampling from the manifold of poses that fulfill orientation constraints. Notably, we show how the IK can be solved in closed form by solving smaller Paden-Kahan subproblems. Secondly, robustness to uncertainties in the orientation data was achieved by introducing a sensor noise model based on the von Mises-Fisher distribution instead of the commonly used Gaussian distribution. Our experiments on diverse complex outdoor video sequences reveal major improvements in the stability and time performance compared to other state-of-the-art trackers. Although in this work we focused on the integration of constraints derived from IMU, the proposed sampling scheme can be used to integrate general kinematic constraints. In future work, we plan to extend our algorithm to integrate additional constraints derived directly from the video data such as body part detections, scene geometry or object interaction.



## References

1. Azad, P., Asfour, T., Dillmann, R.: Robust real-time stereo-based markerless human motion capture. In: Proc. 8th IEEE-RAS Int. Conf. Humanoid Robots (2008)
2. Baak, A., Rosenhahn, B., Müller, M., Seidel, H.P.: Stabilizing motion tracking using retrieved motion priors. In: ICCV (2009)
3. Balan, A.O., Sigal, L., Black, M.J., Davis, J.E., Haussecker, H.W.: Detailed human shape and pose from images. In: CVPR (2007)
4. Bregler, C., Malik, J., Pullen, K.: Twist based acquisition and tracking of animal and human kinematics. *IJCV* 56(3), 179–194 (2004)
5. Chen, J., Kim, M., Wang, Y., Ji, Q.: Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. In: CVPR, pp. 2655–2662. IEEE (2009)
6. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: CVPR, vol. 2, pp. 126–133 (2000)
7. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. *IJCV* 61(2), 185–205 (2005)
8. Fisher, R.: Dispersion on a sphere. *Proceedings of the Royal Society of London. Mathematical and Physical Sciences* (1953)
9. Fontmartry, M., Lerasle, F., Danes, P.: Data fusion within a modified annealed particle filter dedicated to human motion capture. In: IRS (2007)
10. Gall, J., Rosenhahn, B., Brox, T., Seidel, H.P.: Optimization and filtering for human motion capture. *IJCV* 87, 75–92 (2010)
11. Gall, J., Yao, A., Van Gool, L.: 2D Action Recognition Serves 3D Human Pose Estimation. In: Daniilidis, K. (ed.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 425–438. Springer, Heidelberg (2010)
12. Ganapathi, V., Plagemann, C., Thrun, S., Koller, D.: Real time motion capture using a time-of-flight camera. In: CVPR (2010)
13. Gavrilu, D., Davis, L.: 3D model based tracking of humans in action: a multiview approach. In: CVPR (1996)
14. Hartley, R., Zisserman, A.: *Multiple view geometry*, vol. 642. Cambridge University Press, Cambridge (2003)
15. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR, pp. 224–231 (2009)
16. Hauberg, S., Lapuyade, J., Engell-Nørregård, M., Erleben, K., Steenstrup Pedersen, K.: Three Dimensional Monocular Human Motion Analysis in End-Effector Space. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 235–248. Springer, Heidelberg (2009)
17. Kjellström, H., Kragic, D., Black, M.J.: Tracking people interacting with objects. In: CVPR, pp. 747–754 (2010)
18. Lee, C., Elgammal, A.: Coupled visual and kinematic manifold models for tracking. *IJCV* (2010)
19. Lee, M.W., Cohen, I.: Proposal maps driven mcmc for estimating human body pose in static images. In: CVPR, vol. 2 (2004)
20. Lehment, N., Arsic, D., Kaiser, M., Rigoll, G.: Automated pose estimation in 3D point clouds applying annealing particle filters and inverse kinematics on a gpu. In: CVPR Workshop (2010)
21. Moeslund, T., Hilton, A., Krueger, V., Sigal, L. (eds.): *Visual Analysis of Humans: Looking at People*. Springer (2011)
22. Murray, R., Li, Z., Sastry, S.: *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Baton Rouge (1994)

23. Paden, B.: Kinematics and control of robot manipulators. Ph.D. thesis (1985)
24. Pons-Moll, G., Baak, A., Helten, T., Müller, M., Seidel, H.P., Rosenhahn, B.: Multisensor-fusion for 3D full-body human motion capture. In: CVPR, pp. 663–670 (2010)
25. Pons-Moll, G., Rosenhahn, B.: Ball joints for marker-less human motion capture. In: WACV, pp. 1–8 (2009)
26. Pons-Moll, G., Rosenhahn, B.: Model-based pose estimation. In: Visual Analysis of Humans, pp. 139–170 (2011)
27. Pons-Moll, G., Baak, A., Gall, J., Leal-Taixe, L., Mueller, M., Seidel, H.P., Rosenhahn, B.: Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In: IEEE International Conference on Computer Vision (ICCV) (November 2011)
28. Pons-Moll, G., Leal-Taixé, L., Truong, T., Rosenhahn, B.: Efficient and Robust Shape Matching for Model Based Human Motion Capture. In: Mester, R., Felsberg, M. (eds.) DAGM 2011. LNCS, vol. 6835, pp. 416–425. Springer, Heidelberg (2011)
29. Salzmann, M., Urtasun, R.: Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In: CVPR (June 2010)
30. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV, pp. 750–757 (2003)
31. Shoemake, K.: Animating rotation with quaternion curves. ACM SIGGRAPH 19(3), 245–254 (1985)
32. 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)
33. Sigal, L., Balan, L., Black, M.: Combined discriminative and generative articulated pose and non-rigid shape estimation. In: NIPS, pp. 1337–1344 (2008)
34. Sminchisescu, C., Triggs, B.: Kinematic jump processes for monocular 3d human tracking. In: CVPR (2003)
35. Tao, Y., Hu, H., Zhou, H.: Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. IJRR 26(6), 607 (2007)
36. Technologies, X.M.: <http://www.xsens.com/>
37. Urtasun, R., Fleet, D.J., Fua, P.: 3D people tracking with gaussian process dynamical models. In: CVPR (2006)
38. Wang, P., Rehg, J.M.: A modular approach to the analysis and evaluation of particle filters for figure tracking. In: CVPR (2006)
39. Wood, A.: Simulation of the von mises-fisher distribution. Communications in Statistics - Simulation and Computation (1994)
40. Zhang, F., Hancock, E.R., Goodlett, C., Gerig, G.: Probabilistic white matter fiber tracking using particle filtering and von mises-fisher sampling. Medical Image Analysis 13(1), 5–18 (2009)

# On Performance Analysis of Optical Flow Algorithms

Daniel Kondermann<sup>1</sup>, Steffen Abraham<sup>2</sup>, Gabriel Brostow<sup>3</sup>,  
Wolfgang Förstner<sup>4</sup>, Stefan Gehrig<sup>5</sup>, Atsushi Imiya<sup>6</sup>, Bernd Jähne<sup>7</sup>,  
Felix Klose<sup>8</sup>, Marcus Magnor<sup>8</sup>, Helmut Mayer<sup>9</sup>, Rudolf Mester<sup>10,\*</sup>,  
Tomas Pajdla<sup>11</sup>, Ralf Reulke<sup>12</sup>, and Henning Zimmer<sup>13</sup>

<sup>1</sup> Heidelberg Collaboratory for Image Processing  
Interdisciplinary Center for Scientific Computing  
University of Heidelberg

69120 Heidelberg, Germany

[daniel.kondermann@iwr.uni-heidelberg.de](mailto:daniel.kondermann@iwr.uni-heidelberg.de)

<http://hci.iwr.uni-heidelberg.de>

<sup>2</sup> Robert Bosch GmbH, Germany

<sup>3</sup> University College London, United Kingdoms

<sup>4</sup> Bonn University, Germany

<sup>5</sup> Daimler AG, Germany

<sup>6</sup> Chiba University, Japan

<sup>7</sup> Heidelberg University, Germany

<sup>8</sup> Technical University Braunschweig, Germany


<sup>9</sup> Bundeswehr University Munich, Germany

<sup>10</sup> Linköping University (Sweden) and Goethe University, Frankfurt, Germany

<sup>11</sup> Czech Technical University in Prague, Czech Republic

<sup>12</sup> Humboldt University Berlin, Germany

<sup>13</sup> Saarland University, Germany

**Abstract.** Literally thousands of articles on optical flow algorithms have been published in the past thirty years. Only a small subset of the suggested algorithms have been analyzed with respect to their performance. These evaluations were based on black-box tests, mainly yielding information on the average accuracy on test-sequences with ground truth. No theoretically sound justification exists on why this approach meaningfully and/or exhaustively describes the properties of optical flow algorithms. In practice, design choices are often made based on unmotivated criteria or by trial and error. This article is a position paper questioning current methods in performance analysis. Without empirical results, we discuss more rigorous and theoretically sound approaches which could enable scientists and engineers alike to make sufficiently motivated design choices for a given motion estimation task. 

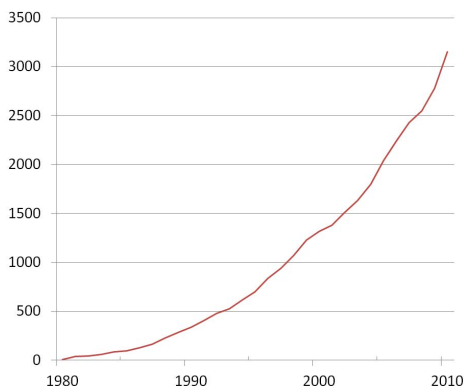
---

\* With support by the Swedish ELLIIT initiative.

<sup>1</sup> This article summarizes the author's results of working group discussions at the Dagstuhl Symposium on Outdoor and Large Scale, Real-World Scene Analysis held in 2011.

## 1 Introduction

The aim of optical flow (OF) algorithms is to compute a motion vector field based on an image sequence (the problem of defining OF properly is discussed in Section 2). OF analysis in image processing and computer vision is a comparatively young field of research with an approximate birthday in the early 80ies [1, 2]. Nonetheless, for more than thirty years, many solutions for OF problems have been proposed: a search on Google Scholar reveals that about every ten years the number of existing publications with the term "optical flow" appearing in the title doubled, reaching around 3000 this year (cf. Figure 1). Among these articles, around 150 have been published in four major journals (IJCV, PAMI, IP, CVIU) since 1980. Counting the number of publications in these journals using the term "optical flow" in the full text, the number for these journals goes up to around 1600.



**Fig. 1.** Cumulative number of publications with optic or optical flow in title based on scholar.google.com (no patents, articles only in the fields "Engineering, Computer Science, and Mathematics", these fields are defined by Google).

A lot of the investigations in these papers deal with the question whether a problem for a specific application can be solved at all with image processing techniques. Today, it seems likely that many interesting problems might be solved using image processing. Although we focus on OF estimation methods, this discussion also relates to other image processing and computer vision methods such as stereo estimation, medical registration, segmentation and denoising.

Yet, with the advent of commercial applications and a ripening field of research, new challenges arise. In this position paper, we specifically discuss the problem of performance analysis which is becoming more and more important in applications such as those involving security risks (e.g. driver assistance systems). We use the term *performance analysis* rather than *benchmarking*, *evaluation* or

<sup>2</sup> Source: [scholar.google.com](http://scholar.google.com), 26.07.2011

*ranking* with the intent to draw attention to the fact that the performance of an algorithm consists of a set of criteria (or requirements) that can vary with the needs of different applications and types of data. As we will discuss, we want to emphasize that performance characteristics of an algorithm cannot be described by a single scalar value.

Starting out from a discussion of contemporary performance analysis approaches in OF problems, we will address each challenge in performance analysis in a separate subsection of this text. Our aim is not to define a new paradigm for performance analysis for OF problems. Neither do the authors offer experimental results on or implementations of existing methods. Instead, the aim of the paper is:

- to review related literature,
- to create awareness for new problems arising due to the increasing number and complexity of existing OF algorithms,
- to show current trends of ongoing discussions among scientists as well as practitioners,
- to propose various new ways to characterize computer vision algorithms,
- and thereby to suggest new fields of research addressing the problems identified in these discussions.

## 1.1 Related Work

Both experimental and theoretical performance analysis of algorithms have a long-standing history in computer science and mathematics (e.g. rooted in complexity theory), whereas system characterization and specification is a similar strand of research in engineering (e.g. requirements analysis in software engineering).

Although many OF algorithms have been suggested, only four publications on their performance analysis exist. Chronologically, the first ones date back to 1994 [3, 4]. At this point around 500 papers with optical flow in their title had been published. In 2001, McCane et al. [5] created a new benchmark, including new synthetic scenes and a free software framework to generate new datasets. The most influential paper was published in 2007 by Simon Baker et al. [6, 7]. The authors not only created new datasets (with extraordinary efforts) and evaluated a new set of algorithms; they also created a website known as Middlebury-Database which has since been used by authors of new OF algorithms to compare their results with others. Today, around forty algorithms have been added to this database. However, compared to the very large corpus of existing work in this field, the number of evaluations is still small and lacks a theoretically justified framework.

The remainder of this section deals with papers on general theoretical approaches to performance analysis in computer vision. In later sections, we will address related work on each of the more specific topics we believe to be relevant for performance analysis of optical flow methods.

In the late 90ies, a number of workshops have been held dealing with performance analysis for computer vision in general [8–10], laying out a roadmap

on why and how this strand of research should and could be established in the community. A general discussion of ten pros and cons for performance analysis in image processing was listed by Förstner [11]. This paper very much reflects the facts that on the one hand performance analysis can be very difficult, expensive and cumbersome, but on the other hand, it is also very important and feasible in terms of longterm research goals. In the same workshop, Maimone and Shafer [12] state six steps necessary for performance analysis: mathematical analysis, simulations without noise, simulations with noise, empirical testing with real data with full control, empirical testing with real data with partial control and empirical testing with uncontrolled data. A year later, these steps have been cited in a workshop editorial by Christensen et al. [8]. In 1998, Matei [13] addressed the first step by suggesting a statistical framework he called "resampling paradigm", whereas Klausmann et al. [14] concentrated on the practical question on how to evaluate performance based on given applications. They were the first to explicitly state that performance characterization and algorithm ranking are two different tasks which should be addressed only if a clear definition of the application of an algorithm is given. Therefore, they define a requirement profile and an assessment function respectively. They argue that: *"The assessment of computer vision algorithms is more than just a question of statistical analysis of algorithm results. Rather, the algorithm field of application has to be taken into account as well."*

In 2001, Courtney and Thacker [15] stated that current research focuses too much on innovation and sophistication and that performance analysis is not carried out in a well-motivated, rigorous manner. They explicitly mention that showing results on a few test images is insufficient, because it does not allow a statistical analysis. They further argue that computer vision should strictly be regarded as a branch of applied statistics. To carry out performance analyses their approach is to distinguish three evaluation types: Technology evaluation (groups of generic algorithms for generic applications), scenario evaluations (specific algorithms for specific applications) and operational evaluations (analysis of the full end user system). In a series of later papers the authors refine these ideas and suggest more concrete methods on computer vision system design [16–18].

Luxen [19] suggests to accumulate large amounts of data such as many views of the same object to achieve low errors. The results can then be used as almost noise-free ground truth. He also suggests to carefully characterize input and output data of computer vision algorithms in order to better understand under which circumstances which output quality can be expected. Similar to [15], Luxen distinguishes four levels of abstraction in computer vision systems design: intentions (e.g. image matching), functions (e.g. least squares fitting), algorithms (e.g. matrix inversion), implementations (concrete code realizing an algorithm). He argues, that each performance characterization can be based on one of these four fields. Hence, both empirical as well as theoretical studies were needed to fully characterize a system. Finally, similar to [12] he distinguishes three types of reference data for real environments: the first type are human annotations (ground truth), the second type is defined by a pair of reference *data*

(without ground truth) as well as reference *code* and the third type is defined by an arbitrary implementation of an algorithm, but predefined reference input. We will discuss the generation of reference data in Section 3.4.

Further discussions on performance analysis in general can be found in [20] and [21]. The authors of [20] argue that the whole system (including all algorithms in a processing chain) need to be understood as one large optimization problem which should be solved based on a very large reference database. In [21] two important aspects are the notion that performance metrics are subject to change over time and that ground truth is very often easy to obtain in case the problem to be solved is on such a high level that humans can simply answer yes/no-questions to create ground truth. The authors also note the interesting fact that currently document analysis [22], face recognition [23] and tracking/surveillance [24] are predominant fields with many and very detailed performance analyses being published.

Most recently, in a book draft [25], Burfoot picked up on the points of [11], but in a much more explicit way. According to the author, *"The weakness of evaluation in computer vision is strongly related to the fact that the field does not conceive of itself as an empirical science. [...] Instead [...], vision researchers see themselves as producing a suite of tools."* (p.103). Burfoot further states: *"A critical reader of the computer vision literature is often struck by the fact that different authors formulate the same problem in very different ways.[...] The cause of this ambiguity in problem definition is that computer vision has no standard formulation or parsimonious justification. [...] Vision papers are often justified by a large number of incompatible ideas. [...] They will also often include completely orthogonal practical justifications, arguing that certain low-level systems will be useful for later, high-level applications."* (p. 104).

He also sees similarities to historical problems in other fields of science such as physics and chemistry: *"It is almost as if, by viewing birds, researchers of an earlier age anticipated the arrival of artificial flight, and proposed to pave the way to that application by developing artificial feathers."*(p. 106) *"The argument of this book, then, is that the conceptual obstacle hindering progress in computer vision is simply a reincarnation of one that so long delayed the development of physics and chemistry."* (p. 108) *"The difference is that physicists can eventually determine which explanation is the best. One crucial aspect of the success of the field of physics is that physicists are able to build on top of their predecessors' work."* (p. 105)

We would like to encourage a discussion on these hypotheses with respect to optical flow estimation. In the remainder of this work we will first review what is actually meant by the term "optical flow" (Section 2). Then, we suggest a number of approaches to consolidate optical flow estimation research in the future.

## 2 Defining Optical Flow

Before we can characterize the properties of an algorithm, we need to clearly define what we mean by "optical flow algorithm". Using the notion of a function

signature in programming, we therefore ask for input and output *datatypes*. Several definitions can be found in textbooks (e.g. [26–28]). According to Burton and Radford [26], the term ”optical flow” is defined as: ”the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.”. This defines the output datatype to a certain degree. Remaining questions are for example whether dense or sparse flow fields need to be found; in case the actual vision system is interested in segmenting an image, the motion contours might be of interest. For tracking applications, the 3D motion of a physical object computed from the flow field could be the output whereas for motion detection, a thresholded flow field might suffice.

The question for the input datatype is more difficult to answer due to several reasons.

First, there often is no notion about the kind of images used as input. Sometimes images come from different spectra (e.g. infrared, x-ray, ...) or optical systems (e.g. fisheye lenses, omnidirectional cameras) and sometimes not all pixels in the image contain useful information (e.g. in the case of particle image velocimetry as defined in [29]). Second, mostly two images are assumed as input, therefore forbidding the use of more than two images in a sequence. Additionally, depending on how strictly this definition is interpreted, it implicitly assumes that there is a bijection, mapping pixel locations in the first image to locations in the second image. Thus, on a discrete grid, occlusions, divergences and convergences are assumed to be negligible, leaving only globally constant translations and rotations as possible outcome of optical flow algorithms. These limitations can be overcome by extending the orthodox notion of optical flow, e.g. by acknowledging and making use of the finite exposure times of images [30].

Of course these definitions are refined or varied in each publication accordingly to describe challenges given a specific application. Usually, all approaches are subsumed under some general term such as optical flow, medical registration, stereo estimation or particle image velocimetry. This is useful to group subsets of OF algorithms with respect to their application domain and typical model assumptions. However, this terminology comes with two disadvantages: first, it is often unclear which application domains are associated with one of these groups. For example, a temporally consistent, non-dense algorithm for pixel-accurate estimation for motion utilizing more than two image frames of a sequence at once can be considered an optical flow algorithm. On the other hand, the algorithm cannot easily be compared by means of the Middlebury database for optical flow evaluation because the number of frames of the test sequences might be too small to yield good results.

The second disadvantage is that it creates the illusion in the mind of the reader, that those algorithms are comparable in their general performance. For example, an algorithm estimating motion in image sequences recorded from inside a car in order to ultimately assist the driver in detecting potential obstacles might be highly similar to an algorithm estimating the motion of a swarm of bees in their nest in order to ultimately understand the communication encoded



in their dance. Yet, each algorithm can be based on completely different assumptions such as there is a planar street on which the camera is moved or that the bees move on a hexagonal grid. The algorithms might also address different problems as for example the occlusion and translucency of cars or motion blur of the bee's shaking bodies. Furthermore, the outcome of the algorithm might be subject to requirements such as sub-pixel accuracy for time-to-impact computation versus good motion boundaries for bee-body segmentation.

Due to these disadvantages of adding all OF algorithms to a single group, we believe that a very careful categorization based on the properties defined in the following sections is crucial for further advancements in the field.

As correspondence problems are mostly ill-posed, prior knowledge about the estimates to be computed is always needed. This knowledge should be well-understood and described as well as possible and also be as accurate as needed for the task. On the other hand, it should generalize well over many types of input data. Therefore, as in machine learning, a trade off between generalization and specialization for the model needs to be found. This condenses to the question: which model is too general and which is too specific? In contrast, current approaches to performance analysis try to categorize existing algorithms either based on the employed optimization framework (e.g. local versus global and variational versus graphical models) or are based on a single scalar output criterion such as the average endpoint error (defined as  $E_{ep}(\mathbf{x}) = \|\mathbf{u}(\mathbf{x}) - \mathbf{g}(\mathbf{x})\|_2$  with  $\mathbf{x}$  being a pixel location,  $\mathbf{u}(\mathbf{x}) = (u_x(\mathbf{x}), u_y(\mathbf{x}))^T$  the computed flow and  $\mathbf{g}(\mathbf{x})$  the true flow, respectively). Yet, instead of being fixed to a single criterion such a ranking needs to take into account all requirements of a given application.

Given a system that uses correspondences as input data (an application), requirements analysis (cf. e.g. [31]) helps to understand how specific the model can be without loss of generalization within the bounds of the application domain. But working with requirements implies knowledge about the application domain. Hence, in order to clearly define OF algorithms we need to create a categorization of applications, which will be discussed now.

## 2.1 Application Categorization/Systematization

In order to analyze the appropriateness of a model for a given application, we need to know the application. On the other hand, there might be an infinite number of yet unknown applications for OF algorithms. It seems unlikely that we can first enumerate all applications and then analyze the performance of each and every algorithm for each and every application. System engineers (cf. e.g. [32]) found a way around this problem by identifying a number of meaningful and intuitive properties for each system component which are measured and then listed in a specification sheet. These properties are selected by finding those which are, ideally, important for as many relevant applications as possible. In order to select the most indicative properties, all currently available applications are considered. Then, by experimentation, system properties are selected and tested for their usefulness.

Currently, the two most important properties for OF algorithms seem to be average endpoint error (disparity error in stereo) and (to some extent) algorithm complexity (computation time, memory efficiency). By looking at some well-known applications we will see that there is a variety of other properties that are important and sometimes contradicting each other.

For photogrammetry and 3D reconstruction [33, 34] correspondences are the basis for triangulations: if the 2D position of the same 3D point is known for two or more views in a number of images, the 3D position can be reconstructed via projective geometry. The accuracy of this reconstruction largely depends on the accuracy of the correspondence (which, in turn, depends on system configuration parameters such as camera baseline, etc.; cf. e.g. [35]). Such methods typically require a large number of correspondences which are not spatially correlated by regularization techniques. The correlation due to these (in general necessary) techniques is a severe problem in statistical analysis as it is difficult to characterize. If the regularization is data dependent or robust estimators are used, the problem becomes even more theoretically involved.

As soon as very large scenes have to be reconstructed, speed and memory efficiency become an issue as well [36]. Here, a tradeoff between speed and accuracy has to be found. This leads to the notion of "scalable algorithms" where an optimum tradeoff can be found by adjusting system parameters.

In robotics and driver assistance systems, OF algorithms have different requirements: in this scenario the task often is to merely detect objects such as traffic signs, the ground plane or sources of danger. Here, speed, memory and energy consumption play a crucial role. On the other hand, sparse flow fields often are sufficient, e.g. for navigation and localization [37, 38].

Correspondences are also used to interpolate intermediate frames between two consecutive time steps of an image sequence [7, 39]. A related case is stereo baseline adjustment or, more general, view synthesis based on multiple images [40]. Software companies involved in cinematic movie postproduction such as The Foundry (Nuke) implement a number of (modified) methods known from literature but are not always published [41, 42]. In these applications the correspondences need not necessarily be physically correct; the most important property often is that they are temporally consistent and can be used to produce results which are pleasing to the eye.

The opposite is the case in scientific measurements. Application scenarios are for example the mensuration of water waves or plant growth in environmental physics [43, 44], estimation of air streams around objects [45], weather-forecasting [46] and the analysis of fluid motion in heart-assist devices [47]. In all these cases, a small endpoint error of the flow vectors has the highest priority, whereas speed often plays a minor role. Furthermore, the confidence (cf. Section 3.2) of each individual measurement needs to be estimated to allow researchers to assess the outcome of each experiment. Interestingly, for these applications, completely parallel fields of research with little overlap to image processing or computer vision have been established [29, 48], bringing up similar concepts

of correspondence estimation, but focusing on different approaches (e.g. block-matching for motion estimation [49]).

A number of other fields of research deal with optical flow, such as action recognition [50], video surveillance [51], video compression [52], video annotation [53], supervision of elderly people [54], swarm analysis of beehives [55] as well as research in zebrafish embryo development [56].

The abundance of existing and possible applications indicates that a complete overview of applications is difficult to define and maintain. On the other hand, based on the requirements of a subset of these applications, a set of more abstract algorithms properties could be found. Similar to specification sheets of electronic system components, we believe that OF algorithms can be described by carefully characterizing input and output data as well as system properties.

Once a definition (or a set of definitions) for algorithms has been found based on applications and their requirements, we would like to understand how well a given algorithm performs. To answer this question, several challenges have to be solved. This will be discussed in the following section.

### 3 Challenges in Performance Analysis

We identified five points to be considered to thoroughly characterize the performance of an algorithm:

- Input data characterization can help to organize typical image sequences into categories with similar properties (Section 3.1).
- Output data characterization should not only evaluate the accuracy of OF methods. Instead, we suggest a list of six important properties of output data (Section 3.2).
- System properties describe the technical aspects of speed and memory consumption as well as modularity and engineerability (Section 3.3).
- The problem of ground truth generation is largely unsolved, but is one of the most crucial as well as difficult aspects of performance analysis for OF algorithms (Section 3.4).
- Finally, well-motivated performance metrics for the comparison of flow fields have to be found (Section 3.5).

Each of these points is carefully motivated in the following subsections. Related work will be discussed along with suggestions how each topic can contribute to a more thorough and theoretically motivated approach to OF performance analysis. In Section 4, we will discuss hypotheses why so few performance analyses for OF are currently available and why a detailed consideration of each of our points could boost both quality and quantity of optical flow research.

#### 3.1 Input Data Characterization

As discussed above, the type of data inserted into an OF algorithm is not always sufficiently described as "image pair".

**Qualitative Characterization.** First steps in input characterization could be to describe the image acquisition process and the content of the scenes for which the algorithm should work. In many specialized publications as for example in medical image registration the mode of data (x-ray, ultrasound, ...) is usually defined clearly. To extend this description of input data it would be beneficial to describe the full imaging setup including sensors, lenses, lens settings (numerical aperture and focal length), light sources (incident angle, physical shape, spectra), surface material properties (reflectance functions), etc. This is usually done in particle image velocimetry where the setups vary largely [29]: in this special case the input data is a 2D image generated laser sheet that visualizes particles. Here, the motion is considered to be truly 2D-dimensional so that apparent flow and physical motion coincide.

Describing and categorizing the acquisition process and the content of the scenes creates awareness for the task the algorithm was made for, but it will often be difficult to exhaustively explore the data when the algorithm is supposed to work well and when not. Another way to solve this challenge might be the analysis of large amounts of input data ideally fully describing inputs which are suitable for the algorithm.

**Quantitative Characterization.** Local feature vectors containing e.g. orientation and scale information could be used to decide whether a given scene is similar enough to yield acceptable results with the OF method at hand. It might be useful if these features were directly related to known critical situations such as occlusions, low amounts of texture, illumination changes or large motions. Also global features describing the image or the scene as a whole and comparing it to sequences with known outcome might characterize input data in a useful way. However, it remains to be studied whether purely local or purely global features can express the full complexity of data sufficiently for a given application.

There are several possibilities to characterize the specific set of image sequences which are addressed by an OF algorithm. First of all, much research has been dedicated to scene descriptors (e.g. GIST is popular approach [57]). Another possibility is to characterize the structure of the (single) images by more or less standard techniques, such as describing the spatial autocovariance function; this can be done compactly by setting up parameterized models, such as separable exponential decay functions. This description should be completed by at least a rough description of the noise variance. A more careful and detailed characterization would include a parameterized description of the optical point spread function as well as the spatial sensor element dimensions (fill factor, or more detailed). The overall characterization of the *discrete* inter-pixel autocovariance results then from convolving the optical and sensor characterization, and the intrinsic autocovariance function of the image, as it would be if the former two influences were neglectable. This intrinsic image autocovariance function corresponds to what is often discussed as the 'natural' and ubiquitous power spectrum of images *per se*, often modeled as an  $1/f$  power spectrum.

The *temporal* characterization should consider the exposure time (which can range between a small fraction of the temporal distance of frames, and the full inter-frame period). More importantly, the temporal characterization should describe the distribution of apparent 2D velocities (or displacements). In the case of certain applications, in particular for driver assistance, this distribution can be significantly different across the image area, and it can also be dependent on some (measurable) external parameters such as camera motion w.r.t. the fixed world coordinate frame. These characterizations do of course not capture the full characteristics of an 'interesting' image sequence, which is structured into differently moving objects, has occlusions, etc., but it is already a very solid basis for optimally designing the derivative operators needed for all differential methods [58] for designing averaging operators (instead of resorting to 'Gaussians') [59], and furthermore to provide useful priors for the entities which are sought.

In an ideal scenario a set of generative input data models (acquired e.g. by machine learning) could be found which can reliably be used to describe the input data the algorithm was made for. As will be discussed in Section 3.2, another intriguing aspect of input data characterization is to identify local regions in a scene where the model cannot be applied to because it is either too specific or unspecific.

### 3.2 Output Data Characterization

As the results of OF methods are used for many different applications, the quality with respect to a given application can be defined with various optimization goals. Hence, next to characterizing input data the same should be done for the resulting flow fields. We will now describe several approaches starting out from very basic characterization techniques such as using example outputs and qualitative evaluations. Then, we will shortly discuss two seldom addressed output data properties, namely robustness with respect to model violations as well as temporal consistency of flow fields. Finally, we will review research on the heavily studied question for accuracy and a currently evolving approach to confidence estimation.

**Example Output.** The most basic and also a very general way to characterize output data is to provide example outputs of the algorithm. This can for example help programmers to check the correctness of a reimplementations of the method at hand. If large amounts of results are available on various kinds of data it can also facilitate the choice of algorithm for a specific application.

**Qualitative Evaluations.** Another basic approach are qualitative evaluations. In creative image processing, aspects such as visualization, rendering and post-processing of videos, the mere beauty of the results can be of major importance. Typical cases are frame interpolation as well as view synthesis. In such cases it might also be possible to "cheat" on the viewer by creating false results which have no noticeable effect on the outcome of the application. These scenarios also allow for psychological tests analyzing whether the viewer is able to find the errors in, or is otherwise affected by the algorithm outcome [60].

**Robustness with Respect to Model Violations.** In safety-relevant applications such as driver assistance and medical systems, the robustness of model and optimization strategy with respect to data violating the model is of great interest. As there is an infinite number of possible model violations it is difficult to devise general tests. One way to describe output data with respect to model violations is to collect large amounts of data containing common model violations, such as motion blur, lens flares, etc. Another closely related question is how fast the results deteriorate if the model is violated. In case the quality degrades gracefully, the algorithm might be better suited for those applications dealing with safety issues.

**Temporal Consistency.** For video processing, the temporal consistency of the algorithm results are often more important than other properties. A test for this consistency could be carried out by systematically varying original data to see how the outcome changes. This is similar to sensitivity analysis in linear models [61] and machine learning approaches. Two recent articles enforcing this property and showing very promising results are [62, 63].

**Accuracy Limits.** There are several ways to test and compare accuracies of OF algorithms. A major problem is how to measure the error because there is an infinite number of options to define an order (or ranking) between two vectors. Hence, each pair of vectors (i.e. ground truth and measured flow vector) first has to be transformed into scalar values in order to be comparable. Next to the regularly used endpoint error [7] various choices exist. One way is to compute the magnitude of both vectors. This is problematic when ground truth vector and measured flow vector are on the one hand equally long but on the other hand point into opposite directions. The magnitude error would still be zero. Another way would be to compute the angle between two vectors which raises the analog problem: The vectors can be of different magnitude. Another problem here is the singularity for vectors of very small magnitude. To weight these two components of magnitude and angle the so-called angular error defined by [64] has been suggested. This error weights both parts of the errors in a nonlinear and unintuitive manner which was not motivated in the original paper (as discussed in [65]). Depending on the application one error measure or another might be favorable, a fact that should be taken into account when stating the accuracy limits of the algorithm.

Once an error measure has been defined, the error distribution needs to be sufficiently motivated. The problem here is, that this distribution actually depends on image data, ground truth and measured flow. For example, testing of the accuracy with a highly textured region that moves at a constant velocity everywhere yields very low errors with most algorithms. If the images were of constant color (one homogeneous region) the results could be completely wrong. The ground truth could also be arbitrary. Hence, testing on a sequence like Yosemite [66] (or any other small set of sequences) does not adequately represent the quality of the algorithm. It just gives a hint that for this type of scene (e.g. highly textured, smooth and mostly small motion in case of the Yosemite sequence) the

algorithm might actually work well. Thus, in some cases algorithms work very well for extremely small motions, sometimes for very large motions. These limits should be well understood and clearly stated.

Furthermore, representing the error distribution only by its mean and variance for a full image is not sufficient, because only the Gaussian distribution can be fully described by these first two moments. As motion estimation errors are far from being Gaussian distributed it might be more helpful to actually visualize the whole distribution (or parts of it) which in turn raises the problem of density estimation. Another option could be to show per-pixel error measures as is done on the Middlebury website.

Finally, it would be helpful if it was known under which circumstances the most accurate results can be achieved by an algorithm. At first this sounds easy to answer: Constant motion through time and much texture certainly is a simple case. Yet, an image of a Gaussian intensity distribution in a 32 bit quantized image might even yield very accurate results for non-constant motions such as a rotation. Furthermore, it is interesting to which degree the results deteriorate with respect to more challenging image data. To the best of our knowledge, this aspect has never been studied thoroughly although it is very important for scientific applications where sub pixel accuracy is critical and where it is safe to make more specific assumptions about the model.

**Estimability, Confidence and Alternative Solutions.** Usually, OF algorithms are analyzed by comparing ground truth with actual algorithm results. This type of performance analysis is carried out by humans prior to the actual usage of the algorithm in a full computer vision system. Therefore, we call this technique *supervised performance analysis*. An alternative approach is to allow for self-diagnosis of the computer vision system while it is running in its real environment. We call this approach *unsupervised performance analysis* which will be described now.

To motivate three aspects of unsupervised performance analysis consider the following extreme example of OF input data: A typical image sequence for particle velocimetry consists of a mainly black background and some hundred (or thousand) bright moving spots which are physical tracer particles in a fluid. In the black (homogeneous) regions of the background no motion can be estimated: a black spot at any location can be matched to almost any other location in the next frame. We do not care about these occlusions and ambiguities in the background and simply assume that there is no motion at all. Hence, an algorithm working on this data should be able to decide where motion can (or should) be estimated *at all*. Furthermore, particle velocimetry is often used in environmental sciences to measure fluid motion, so each and every measurement must come with (at least) an error bar, showing the *precision* of each flow vector. Finally, occlusion occurs whenever two particles are crossing due to the projective nature of the image acquisition. Sometimes, it is impossible to decide which particle is which after they crossed in the image plane. Therefore, our algorithm needs to be aware of *alternative solutions*.

More generally, we ask how much information we *need* to obtain from the given data and how much we *can* obtain depending on the intended later use of the resulting motion:

- Dealing with occlusions and ambiguities can be understood as dealing with estimatibility: Instead of assuming that at each pixel of an image sequence a full flow can be estimated, we pose the question whether motion can be estimated at all and, if so, how many parameters of it [67, 68]. This should be easier to decide than to actually carry out the estimation.
- To answer the question how accurate the results are we use confidence measures. This should still be easier than computing an actual flow field.
- Finally, the most algorithmically complex and related task would be to not only find one motion estimate but to also inform the user about alternative solutions.

These notions of estimatibility, confidence and alternative solutions also relax the problem of motion estimation: we do no longer need to estimate flows at each and every pixel. This reduces both computational cost and potentially harmful results in safety-relevant applications such as driver assistance systems.

While little literature focuses on estimatibility and alternative solutions for optical flow, confidence measures have already been studied by Barron et al. [3]. A first paper specifically dedicated to the comparison of confidence estimation approaches has been published by Bainbridge and Lane in 1996 [69].

Two approaches are regularly being studied: confidence based on input data (images) and confidence based on output data (flows). As the first does not take the results into account, they can also be interpreted as estimatibility measures. A central theme recurring in all image-based confidence methods is the notion of the local shape of the energy to be minimized. The intuition is that sharp peaks in the energy indicate high confidence whereas low curvatures allow for many equally likely flows resulting in a low confidence.

More formally, two highly related theoretical frameworks can be used to describe this approach: intrinsic dimensions and Fisher information (both defined e.g. in [70]). Both definitions are based on the local covariance matrix of the energy of an OF model. Intrinsic dimensions can for example be used to determine the number of parameters which can be estimated [67]. They have firstly been applied in computer vision in 1990 [71] and later been adopted e.g. in [27] and [72, 73]. Fisher information is used to describe the Cramér-Rao Lower Bound which states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information. Therefore, this bound is an indicator of how accurate the best possible outcome of the motion estimate can be. Another option is to use the Chi-Square-Test which can be used to verify the appropriateness of a model in case the errors are normally distributed, unbiased and have a given assumed covariance matrix.

A different way to estimate confidences is to solely rely on prior knowledge on flow field statistics. This has been studied for example in [74, 75], where the spatio-temporal statistics of typical flow fields are learned in terms of a linear model which is then used to employ hypothesis testing on OF algorithm results.



Similar approaches on learning the statistics of flow fields have previously been applied to OF estimation (e.g. in [76–78]).

Two recent publications [79, 80] use learning based on multiple clues derived from both image and flow data for confidence estimation.

Finally, scene-inherent redundancy could be another aspect for confidence estimation: in case one has three or more images, the results should be consistent with respect to the geometry of the scene, e.g. rays to the same scene points should intersect. This goes beyond the Fisher information, as additional flow fields of other pairs of images of a static scene can be used to define a local flow vector quality criterion.

### 3.3 System Properties

Until now we have focused on the algorithm definition as well as the input and output data characteristics. All these properties focus on the data an algorithm receives and computes. Another important point is to understand all relevant technical details of concrete implementations. Hence, a set of system properties needs to be found so that engineers can deal with a system to compute flow fields as black box. We identified three major groups of such properties: the ease of maintenance and implementation, the possibility of white-box testing and speed as well as memory usage.

**Engineerability and Number of Parameters.** We understand engineerability as the ease of implementation, the possibility to actually implement the algorithm in a commercial application and the possibilities of adapting the method to the specific needs of engineers. Especially the number of parameters influencing the output of the algorithm should be small in their number, intuitive to understand and insensitive with respect to input data. In case the number of parameters cannot easily be reduced, a set of default values should be known which can be used to create results of reasonable quality on most images. Commercial aspects such as whether the algorithm is patented or not might also play a role. This system property can be tested easily by explaining and motivating the parameters thoroughly and estimating the amount of time a programmer new to the field might need to implement the method.

**Modularity and White Box Testing.** A common practice in the publication of OF algorithms is to describe the whole algorithm and to test its output against test sequences. Regularly, a few crucial parts of the algorithm are either left out or parameterized differently in order to estimate its effect on the overall results. For example, many OF algorithms are built up from many algorithmic elements, such as multiple similarity measures, image derivative kernels, interpolation techniques, pyramid computation schemes, regularization terms and so on. Each of these elements has parameters and can even be replaced by completely different methods. For example, sub pixel image intensities can be interpolated by a number of interpolation schemes; an image pyramid can be computed by scaling the original image down by a factor of two or smaller or it can even scale

the image up to some degree [81]; the derivative of an image can be computed by many kernels or even other filtering techniques ranging from simple central differences to sophisticated filters specially designed to estimate motion with a specific similarity measure [82]. Any subtle change in these settings can influence the overall accuracy of the results and is therefore worth further investigation.

At the core of this problem lies the fact that any OF algorithm is actually plugged together from a large set of modules available. Some of these modules as for example image derivative computation are fields of research on their own. It would be helpful if there were a set of known slots (constituting the elements of the most general motion estimation algorithm and clearly defining input and output data) and a variety of possible modules that could be plugged into each appropriate slot. Then, each slot or module could be scientifically investigated separately and also in its combination with other modules (white box testing). A software framework for this approach including a number of example optical flow algorithms has recently been made publicly available<sup>3</sup>. The software is based on a modularization strategy specifically designed for OF algorithms as suggested in [83]. These modules of an optical flow method are another interesting set of algorithm properties.

**Execution Speed and Memory Usage.** The time and memory an algorithm needs to actually estimate the motion of an image sequence usually is a major issue in industrial applications. Several aspects range from practical over completely theoretical to technically highly intricate considerations; to each of these a complete field of research is dedicated. Therefore, it is very difficult to judge the execution speed of an algorithm even though it is one of its important properties.

- Data Reduction: Sometimes, it suffices to only compute motion at a few locations. Hence, computation time can be saved by finding algorithms that reduce the number of locations. This is a typical approach in tracking [84] where usually only very few pixels of an image sequence are investigated.
- Mathematics: For example in global motion estimation techniques (often including systems of partial differential equations), large linear systems of equations are generated from the image sequence. Their solution can be carried out by many methods, ranging from Gaussian Elimination Schemes over Krylov Subspace Methods to Algebraic Multigrid Schemes. Exploiting mathematical properties can dramatically reduce computation times. This was for example shown by [85, 86].
- Parallelization: With the dawn of multicore desktop computers and general purpose GPUs, parallelization has become a major topic. Especially in image processing, parallelization is surprisingly easy to implement (consider e.g. the convolution of an image with a mask). But also solving large linear systems of equations can be done in parallel (cf. e.g. [87]).
- Code Optimization: It might sound trivial but with a diversity of large image processing libraries for major programming languages (as e.g. C++ and

---

<sup>3</sup> <http://charon-suite.sourceforge.net>

Matlab) code optimization is far from simple. Nonetheless, this part can also affect theoretical considerations: if it were for example easier to optimize code for matrices than for other data structures such as graphs, the choice of the optimization method would interfere with the actual code design. Today, a programmer needs to have a deeper understanding on how image processing libraries implement their functionality in order to optimally exploit its internal structures. Another problem is that the ways compilers optimize code is rather unintuitive: one cannot implement all functions in the same way to yield the same automatic code optimizations. A typical approach is trial and error, but each compiler optimizes its code differently so that the same code can be much faster when compiled with a different compiler.

- Choice of Hardware: For some methods, specifically designed hardware ranging from image acquisition device to integrated circuits for numerical optimization can influence the execution speed. For example, modern driver assistance systems contain integrated modules for stereo estimation which deliver highly accurate depth maps in real time with very low power consumption. Another example are highly optimized detectors in the large hadron collider which can detect and transfer collisions in the gigabyte-range per second. Finally, the famous Microsoft Kinect creates depth maps in real time with a customized hardware setup for structured light. This shows that a focus on regular personal computers is not necessarily the best way to decide whether an algorithm can be fast or whether some specific problem can be considered as solved.

Hence, investigations into the various complexities of optical flow algorithms are an important property to be specified.

### 3.4 Ground Truth Generation

The typical approach to evaluate the quality of output data is to design ground truth image sequences where the motion is known. Two approaches can be chosen:

1. Synthetic image sequences are generated. Due to the underlying and known 3D models, the true motion field is generated easily from animation data. The problem with this approach is that rendered images can be unrealistic. In fact, it is unknown whether renderings are realistic enough to fake real-world scenes.
2. Real images are recorded. The motion is measured by some technique which is more accurate than optical flow methods. The problem with this approach is that the measurement motion can be inaccurate and that very few accurate motion estimation techniques are known. This leads to scenes with limited content such as scenes with rigid body transformations, small sets of a collection of rather artificial items and the like.

The dilemma in ground truth generation therefore is that either the ground truth flow fields are too inaccurate or the recorded image sequences are too artificial.

The most famous examples for synthetic scenes are the Yosemite sequence [66], the street and office sequences [5] and the diverging tree sequence [3]. Of course, they do not cover all types of applications and can therefore only be used as a hint on how the algorithm might perform on other sequences. One problem with such sequences is that it is largely unknown whether they represent important or typical cases of motion together with the rendered images. Furthermore, there are sequences which are acquired with a real camera. The first well-known example is the marbled block sequence [4] which contains a few block-shaped, textured objects standing on a textured underground. Recently, a number of new synthetic and real sequences have been generated by [7]. Furthermore, the authors of [7] encourage the publication of results based on a website where everyone can submit new motion fields. For automotive scenarios three large datasets have been published [88–90]. They both contain very large amounts of representative data, but for [89], no ground truth is available whereas [88, 90] partly have been augmented with ground truth.

Furthermore, the generation of ground truth data is a challenging optical measurement task itself. Its accuracy should ideally be magnitudes above the accuracy that can be achieved by motion estimation algorithms. The typical problem of real sequences is the estimation of this accuracy. In the publications mentioned above the information supplied from an optical measurement perspective seems to be insufficient to clearly state accuracy limits. Hence, even though in real sequences all physical imaging effects from lens distortion and noise to light reflections and refractions are modeled properly, it remains unclear whether their ground truth is good enough. In such circumstances, when ground truth of real world data is either difficult or impossible to obtain, one can either use human-assisted motion annotations [91, 92] and carefully evaluate the accuracy of the resulting flow fields or one can try to synthetically create image sequences with known ground truth. One tool to achieve the latter has recently been suggested by [79].

Then, an open question is whether rendered scenes are sufficient to simulate the real world with respect to OF methods. Inspired by a first analysis of real versus synthetic data [88], in [93, 94] the goal was to create the same scene both in the computer and in reality and to compare the outcome of a given OF algorithm. In case the two results do not differ significantly, we can conclude that computer graphics can be used to simulate at least a part of reality. How large this part is would then be subject of further investigations. Yet, along with [11], we would like to stress the point that simulated data are absolutely necessary to prove the correctness and potential accuracy of algorithms.

Finally, the selection of the (ideally) *best* datasets is a big and completely unsolved problem. In practical applications, we are required to evaluate OF without ground truth. Therefore, we need to believe that the results computed by algorithms which derive acceptable performance for reference data are also acceptable for other sequences. However, to accept this meta-criterion, we are required to accept the pre-assumption that mathematical, geometrical, and physical properties of the test data are at least comparable to the previously used

reference datasets. Therefore, we need to evaluate the quality of our datasets: which scenes do represent real data best for a given application?

In current real datasets, the camera often does not move. In synthetic datasets the camera is often flying smoothly through the scene. Both types of camera motion seem unlikely in real-world situations such as robotics or driver assistance systems. On the other hand, in surveillance applications a static camera can very often be assumed, whereas in airborne settings, a smoothly flying camera might be a good assumption. Next to the camera motion, the content of the scene and the motions present in the scene have to be decided on some well-motivated thoughts. For example, most probably nobody wants to estimate the motion of fireworks exploding in a breaking ocean wave during a blizzard with big snow flakes and lightning bolts. On the other hand, difficult sequences such as the motion analysis of a soccer game during rain with hundreds of strobe lights triggered by reporters can be highly valuable.

Another problem for the best selection of sequences with realistic camera and object motions is the length of the sequence: In case motion is temporally coherent in our reference datasets, we can use the computed results of the previous frame as a guide to evaluate the results in the present frame. Yet, this property implies that to evaluate algorithms which will be applied to long image sequences, we are required to prepare reference image sequences which satisfy the same temporal motion coherence along the time axis.

Little related work on this topic exists; a first step towards the question of good datasets was proposed in [95]. Outside the field of OF, Shotton et al. showed that for human pose estimation it is feasible to build a challenging synthetic test (and training) dataset [96]. Kaneva et al. used this idea for feature estimation [97].

Even if ground truth data could be easily generated in large amounts, it would still be unclear whether a generalization of the image data created across all fields or even inside each field of applications can be found. Thus, the quality assessment of something like a general-purpose optical flow algorithm might still be impossible: We would have to test it with all types of test sequences we can imagine. Therefore, even if a general-purpose algorithm were found, we would possibly never be able to identify it. We argue that to alleviate this problem many more sequences need to be created. If a generative model for input characterization methods (as discussed in Section 3.1) would be found, one way to use it would be to generate such large amounts of ground truth. As optical flow scientists usually have a specific type of images in mind, another way to alleviate this problem is to supply the source code of the algorithms in order to enable other scientists to carry out tests with their own data. This method may seem obvious but is, unfortunately, not always put into practice.

### 3.5 Performance Metrics

New motion estimation algorithms are usually tested with a number of ground truth sequences. Until the Middlebury Database was established in 2007, they were often solely tested against the Yosemite sequence [66]. As an error measure usually the so called average angular error (defined by [64] and used by [3

and most successive papers) and its standard deviation over a single frame of this sequence is reported. Not only is this error measure unmotivated, it also is inappropriate for the comparison of some typical problems of motion estimation as is e.g. laid out in [27]. To address these problems, an additional set of performance measures was introduced by [7]. But it is not obvious which measure can best be used to compare the estimated results to the ground truth.

To put it in a nutshell, currently used performance measures are of questionable use in real-world application scenarios. A lot of future research could be carried out in this field.

## 4 Conclusion and Future Research

In this paper we have discussed the importance of performance analysis for optical flow algorithms. A number of algorithm characteristics have been proposed to help scientists as well as engineers to design improved algorithms or choose between several options for a given application.

How can we facilitate systematic performance analyses of existing OF algorithms?

In the past much attention was paid to *innovation* of new methods rather than *consolidation* of existing methods. This resulted in an abundance of publications. To better understand these findings in OF research, we suggest the following first steps to consolidate existing work.

### 4.1 Creation of Reference Implementations

Creating a new implementation of existing methods is a time-consuming task due to the increasing complexity in current modeling and optimization techniques. Often, much theory knowledge and programming expertise are needed. Yet, it would help to have multiple independent implementations of each OF method for performance analysis.

Implementing an existing method could be rewarded by scientific reputation: the online journal Image Processing On Line is dedicated to certifying algorithm implementations<sup>4</sup>, so that peer-reviewed implementations of OF methods become part of a scientific result. This approach has many advantages:

- Peer-reviewed reference implementations would be generally accepted by the community.
- Comparisons to baseline methods became possible without ambiguity due to implementation details.
- The workload of reimplementing existing methods is distributed over the community.
- Performance analyses of new methods become easier.

For future research, we encourage students and scientists to publish peer-reviewed reference implementations to create a basis for consolidation in OF research.

---

<sup>4</sup> [www.ipol.im](http://www.ipol.im)

## 4.2 Creation of a System Characterization Standard

We have suggested a number of ways to characterize OF algorithms. We showed that, next to accuracy, speed and innovations in modeling, there are many interesting properties. Characterizing them could lead to new approaches with very good tradeoffs for the specialization-generalization-dilemma stated above. This article is a step towards more awareness for system characterization in OF. Further position papers, workshops or even dedicated journals or conferences could help to create a system characterization standard which is supported by a majority of researchers.

## 4.3 Specialization of Publications on Subtopics in OF

Historically, publishing a new paper in OF is done by reviewing the related work and describing a model as well as optimization technique. Experiments are shown indicating that the proposed method works well under reasonable assumptions. In the nineties, the number of publications was already so large that it became difficult to exhaustively describe the related work. The first review papers emerged and authors of new methods concentrated on the closest related work in order to be able to keep the page limit.

Today, models and optimization techniques become more and more sophisticated and the number of OF publications has grown out of the bounds of an exhaustive review paper. Additionally, performance analysis has become more important as engineers need to choose from among thousands of publications "the correct" method for their specific application. As a result, it became difficult to give all answers about a new approach within a single publication.

Breaking down the OF problem into parts which can be handled conveniently and in great detail within a single article could therefore be beneficial. One approach could be to only propose a new model in a baseline optimization framework and show that the results make sense (but without performance analysis) and the idea is innovative. Other researchers could create and/or use a reference implementation to study its properties as proposed in this article. Yet another group could compare the results with those of other methods. Finally, a paper about many comparisons could come to a conclusion about the question which method is most appropriate for which task.

Thus, innovation and consolidation and could be significantly facilitated.

## 4.4 Usage of White-Box-Testing for Performance Analysis

Black-box-testing analyzes the properties of an OF algorithm solely based on its output [98]. The advantages of this approach are that no knowledge about the internals is necessary and users will experience the same behavior. A disadvantage is that it remains unclear which component of the method caused a change of the system properties: for example, exchanging a scheme for pyramid or derivative computation can have a large impact on the output.

Whenever multiple modules are modified at the same time, black-box-testing is no longer suitable to interpret the results: it might be possible that two of three modified modules degrade the outcome whereas the third modules yields a very significant improvement. If several results from more than one publication are to be compared, we simply cannot change one module at a time.

These are reasons to use so-called white-box-testing, meaning that the effect of each module of an algorithm on the system properties should be analyzed separately. One approach is to segment OF algorithms into independent modules and create reference implementations for each module separately. Algorithms sharing several modules such as pyramid or derivative computation can then be easily compared. This approach has been described in [83] and resulted in a freely available, modular software suite called Charon<sup>5</sup>.

#### 4.5 Development of a Simple Ground Truth Generation Technique

Categorizing OF applications and finding a way to characterize input and output data as prerequisites for thorough performance analyses is a difficult task in its own right. Until a standard approach has been found it would still be useful to evaluate OF algorithms with respect to specific applications. Creating many ground truth sequences is a good way to achieve this, but as the number of applications is very large it is difficult to create so many sequences. Ideally, everybody should be able to easily create new ground truth satisfying some well-defined quality constraints. Possible candidates for such an approach would be synthetic image sequences or 3D scanning. Both ideas need a sound scientific validation before they can be employed as a black box.

#### 4.6 Summary

We have suggested five directions for future research: reference implementations, system characterization standards, subtopics for publications, white-box-testing and simple ground truth generation. A better balance between consolidation and innovation could be found by these approaches. With this article we hope to inspire scientists to have a closer look at what has already been achieved in our field of research.

## References

1. Horn, B., Schunck, B.: Determining optical flow. In: Artificial Intelligence, vol. 17, pp. 185–204 (1981)
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 1981 DARPA Image Understanding Workshop, pp. 121–130 (1981)
3. Barron, J.L., Fleet, D.J., Beauchemin, S.: Performance of optical flow techniques. International Journal of Computer Vision 12(1), 43–77 (1994)

---

<sup>5</sup> [charon-suite.sourceforge.net](http://charon-suite.sourceforge.net)



4. Otte, M., Nagel, H.: Optical Flow Estimation: Advances and Comparisons. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 51–60. Springer, Heidelberg (1994)
5. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking optical flow (2001), <http://of-eval.sourceforge.net/>
6. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: Proc. of the 11th International Conference of Computer Vision (ICCV 2007), pp. 1–8. IEEE (2007)
7. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. *International Journal of Computer Vision* 92(1), 1–31 (2011)
8. Christensen, H., Förstner, W.: Editorial performance characteristics of vision algorithms. *Machine Vision and Applications* 9(5), 215–218 (1997)
9. Haralick, R., Klette, R., Stiehl, S., Viergever, M.: Performance characterization in computer vision (2000)
10. Clark, A., Courtney, P. (eds.): ICVS workshop on performance characterization and benchmarking of vision systems (1999)
11. Förstner, W.: 10 pros and cons against performance characterization of vision algorithms. In: Proc. of ECCV Workshop on Performance Characteristics of Vision Algorithms, pp. 13–29 (1996)
12. Maimone, M., Shafer, S.: A taxonomy for stereo computer vision experiments. In: Proc. of ECCV Workshop on Performance Characteristics of Vision Algorithms, pp. 59–79 (April 1996)
13. Matei, B., Meer, P., Tyler, D.: Performance assessment by resampling: rigid motion estimators. In: Proc. IEEE CS Workshop on Empirical Evaluation of Computer Vision Algorithms, Santa Barbara, California, pp. 72–95 (1998)
14. Klausmann, P., Fries, S., Willersinn, D., Stilla, U., Thönnessen, U.: Application-oriented assessment of computer vision algorithms. In: *Handbook of Computer Vision and Applications*, vol. 3, pp. 133–152 (1999)
15. Courtney, P., Thacker, N.: Performance characterisation in computer vision: The role of statistics in testing and design. In: *Imaging and Vision Systems: Theory, Assessment and Applications*. NOVA Science Books (2001)
16. Thacker, N., Lacey, A., Courtney, P., Rees, G.: An empirical design methodology for the construction of machine vision systems. In: Tutorial at ECCV, Copenhagen (2002)
17. Thacker, N.: Using quantitative statistics for the construction of machine vision systems. In: *Proceedings of SPIE: Opto-Ireland 2002: Optical Metrology, Imaging, and Machine Vision*, vol. 4877, pp. 1–15 (2003)
18. Thacker, N., Clark, A., Barron, J., Ross Beveridge, J., Courtney, P., Crum, W., Ramesh, V., Clark, C.: Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding* 109(3), 305–334 (2008)
19. Luxen, M.: Performance evaluation in natural and controlled environments applied to feature extraction procedures. In: Proc. of 2004 ISPRS Congress. *The International Archives of The Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXV, Part B3, pp. 1061–1066 (2004)
20. Lucas, Y., Domingues, A., Driouchi, D., Treuillet, S.: Design of experiments for performance evaluation and parameter tuning of a road image processing chain. *EURASIP Journal on Applied Signal Processing*, 212 (2006)
21. Vogel, J., Schiele, B.: On Performance Characterization and Optimization for Image Retrieval. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 49–63. Springer, Heidelberg (2002)

22. Liang, J., Doermann, D., Li, H.: Camera-based analysis of text and documents: a survey. *International Journal on Document Analysis and Recognition* 7(2), 84–104 (2005)
23. Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: Face recognition: A literature survey. *ACM Computing Surveys (CSUR)* 35(4), 399–458 (2003)
24. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. *ACM Computing Surveys (CSUR)* 38(4), 13 (2006)
25. Burfoot, D.: Notes on a new philosophy of empirical science. Arxiv preprint arXiv:1104.5466 (2011)
26. Burton, A., Radford, J.: *Thinking in perspective: critical essays in the study of thought processes*. Methuen (1978)
27. Haussecker, H., Spies, H.: Motion. In: Jähne, B., Haussecker, H., Geissler, P. (eds.) *Handbook of Computer Vision and Applications*, vol. 2, ch. 13. Academic Press (1999)
28. Warren, D., Strelow, E.: *Electronic spatial sensing for the blind: contributions from perception, rehabilitation, and computer vision*, vol. 99. Kluwer Academic Print on Demand (1985)
29. Raffel, M., Willert, C., Kompenhans, J.: Postprocessing of PIV data. In: *Particle Image Velocimetry*, ch. 6. Springer (1998)
30. Sellent, A., Eisemann, M., Magnor, M.: Two Algorithms for Motion Estimation from Alternate Exposure Images. In: Cremers, D., Magnor, M., Oswald, M.R., Zelnik-Manor, L. (eds.) *Video Processing and Computational Video*. LNCS, vol. 7082, pp. 25–51. Springer, Heidelberg (2011)
31. Maciaszek, L.: *Requirements analysis and system design*. Pearson Education (2007)
32. Kossiakoff, A., Sweet, W., Seymour, S., Biemer, S.: *Systems engineering principles and practice*, vol. 27. Wiley Online Library (2003)
33. Mikhail, E., Bethel, J., McGlone, J.: *Introduction to modern photogrammetry*, vol. 31. Wiley, New York (2001)
34. Hartley, R.I., Zisserman, A.: *Multiple View Geometry*. Cambridge University Press (2000)
35. Thormaehlen, T.: *Zuverlässige schätzung der kamerabewegung aus einer bildfolge* (2006)
36. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part IV*. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010)
37. Guilherme, N., Avinash, C.: Vision for mobile robot navigation: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(2), 237–267 (2002)
38. Ohnishi, N., Imiya, A.: Featureless robot navigation using optical flow. *Connection Science* 17(1-2), 23–46 (2005)
39. Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics (TOG)* 23, 600–608 (2004)
40. Chen, S., Williams, L.: View interpolation for image synthesis. In: *Proc. of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 279–288. ACM (1993)
41. Parsonage, P., Hilton, A., Starck, J.: Efficient dense reconstruction from video. In: *Proceedings of the 8th European Conference on Visual Media Production* (2011), <http://www.cvmp-conference.org/2011-Papers>

42. Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., Gross, M.: Non-linear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics (TOG)* 29(4), 75 (2010)
43. Garbe, C., Jähne, B.: Reliable estimates of the sea surface heat flux from image sequences. *Pattern Recognition*, 194–201 (2001)
44. Barron, J., Liptay, A.: Measuring 3-d plant growth using optical flow. *Bioimaging* 5(2), 82–86 (1997)
45. Kähler, C., Sammler, B., Kompenhans, J.: Generation and control of tracer particles for optical flow investigations in air. *Experiments in Fluids* 33(6), 736–742 (2002)
46. Papadakis, N., Mémin, É., et al.: Variational assimilation of fluid motion from image sequence. *SIAM Journal on Imaging Science* 1(4), 343–363 (2008)
47. Berthe, A., Kondermann, D., Christensen, C., Goubergrits, L., Garbe, C., Affeld, K., Kertzscher, U.: Three-dimensional, three-component wall-PIV. *Experiments in Fluids* 48, 983–997 (2010)
48. Tropea, C., Yarin, A.L., Foss, J.F.: *Springer Handbook of Experimental Fluid Mechanics*. Springer (2007)
49. Fincham, A., Spedding, G.: Low cost, high resolution dpiv for measurement of turbulent fluid flow. *Experiments in Fluids* 23(6), 449–462 (1997)
50. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: *Proc. of the 8th International Conference of Computer Vision (ICCV 2003)*, pp. 726–733. IEEE (2003)
51. Haag, M., Nagel, H.: Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision* 35(3), 295–319 (1999)
52. Wolf, W.: Key frame selection by motion analysis. In: *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1996)*, vol. 2, pp. 1228–1231. IEEE (1996)
53. Sudhir, G., Lee, J.: Video annotation by motion interpretation using optical flow streams (1997)
54. Hauptmann, A., Gao, J., Yan, R., Qi, Y., Yang, J., Wactlar, H.: Automated analysis of nursing home observations. *IEEE Pervasive Computing* 3(2), 15–21 (2004)
55. Michels, M., Rojas, R., Landgraf, T.: A beehive monitoring system incorporating optical flow as a source of information (2011)
56. Lombardot, B., Luengo-Oroz, M., Melani, C., Faure, E., Santos, A., Peyrieras, N., Ledesma-Carbayo, M., Bourguine, P., de Neurobiologie Alfred Fessard, G., Yvette, F.: Evaluation of four 3d non rigid registration methods applied to early zebrafish development sequences. In: *MIAAB MICCAI* (2008)
57. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research* 155, 23–36 (2006)
58. Krajssek, K., Mester, R.: Wiener-Optimized Discrete Filters for Differential Motion Estimation. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) *IWCM 2004*. LNCS, vol. 3417, pp. 30–41. Springer, Heidelberg (2007)
59. Krajssek, K., Mester, R., Scharr, H.: Statistically Optimal Averaging for Image Restoration and Optical Flow Estimation. In: Rigoll, G. (ed.) *DAGM 2008*. LNCS, vol. 5096, pp. 466–475. Springer, Heidelberg (2008)
60. Stich, T., Linz, C., Wallraven, C., Cunningham, D., Magnor, M.: Perception-motivated interpolation of image sequences. *ACM Transactions on Applied Perception (TAP)* 8, 1–25 (2011), <http://doi.acm.org/10.1145/1870076.1870079>

61. Forstner, W.: Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. *Computer Vision, Graphics, and Image Processing* 40(3), 273–310 (1987)
62. Volz, S., Bruhn, A., Valgaerts, L., Zimmer, H.: Modeling temporal coherence for optical flow. In: *Proc. of the 13th International Conference of Computer Vision, ICCV 2011* (2011)
63. Becker, F., Lenzen, F., Kappes, J.H., Schnörr, C.: Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences. In: *Proc. of the 13th International Conference of Computer Vision, ICCV 2011* (2011)
64. Fleet, D.J., Jepson, A.: Computation of component image velocity from local phase information. *International Journal on Computer Vision* 5(1), 77–104 (1990)
65. Jähne, B., Haussecker, H., Geißler, P.E.: *Handbook of Computer Vision and Application*, vol. 2. Academic Press (1999)
66. Heeger, D.: Model for the extraction of image flow. *Journal of the Optical Society of America* 4(8), 1455–1471 (1987)
67. Kondermann, C., Mester, R., Garbe, C.: A Statistical Confidence Measure for Optical Flows. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III. LNCS*, vol. 5304, pp. 290–301. Springer, Heidelberg (2008)
68. Humayun, A., Mac Aodha, O., Brostow, G.: Learning to find occlusion regions. In: *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2161–2168. IEEE (2011)
69. Bainbridge-Smith, R., Lane, A.: Measuring confidence in optical flow estimation. *IET Electronics Letters* 32(10), 882–884 (1996)
70. Bishop, C.: *Neural Networks for Pattern Recognition*. Oxford University Press, New York (1995)
71. Zetsche, C., Barth, E.: Fundamental limits of linear filters in the visual processing of two-dimensional signals. *Vision Research* 30(7), 1111–1117 (1990)
72. Kalkan, S., Calow, D., Felsberg, M., Worgotter, F., Lappe, M., Krüger, N.: Optic flow statistics and intrinsic dimensionality (2004)
73. Felsberg, M., Kalkan, S., Krüger, N.: Continuous dimensionality characterization of image structures. *Image and Vision Computing* 27(6), 628–636 (2009)
74. Kondermann, C., Kondermann, D., Garbe, C.S.: Postprocessing of Optical Flows Via Surface Measures and Motion Inpainting. In: Rigoll, G. (ed.) *DAGM 2008. LNCS*, vol. 5096, pp. 355–364. Springer, Heidelberg (2008)
75. Kybic, J., Nieuwenhuis, C.: Bootstrap optical flow confidence and uncertainty measure. *Computer Vision and Image Understanding* 115(10), 1449–1462 (2011)
76. Black, M., Yacoob, Y., Jepson, A., Fleet, D.: Learning parameterized models of image motion. In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1997)*, pp. 561–567 (1997)
77. Roth, S., Black, M.: On the spatial statistics of optical flow. In: *Proc. of International Conference on Computer Vision (ICCV 2005)*, vol. 1, pp. 42–49 (2005)
78. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning Optical Flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III. LNCS*, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
79. Mac Aodha, O., Brostow, G.J., Pollefeys, M.: Segmenting video into classes of algorithm-suitability. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pp. 1054–1061 (2010)
80. Gehrig, S., Scharwächter, T.: A real-time multi-cue framework for determining optical flow confidence. In: *Proc. of the 13th International Conference of Computer Vision, ICCV 2011* (2011)

81. Amiaz, T., Lubetzky, E., Kiryati, N.: Coarse to over-fine optical flow estimation. *Pattern Recogn.* 40(9) (2007)
82. Scharr, H.: Optimal Filters for Extended Optical Flow. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) *IWCM 2004*. LNCS, vol. 3417, pp. 14–29. Springer, Heidelberg (2007)
83. Kondermann, D.: *Modular Optical Flow Estimation with Applications to Fluid Dynamics*. PhD thesis, University of Heidelberg (2009)
84. Blackman, S., Popoli, R.: *Design and Analysis of Modern Tracking Systems*. Artech House (1999)
85. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Real-Time Optic Flow Computation with Variational Methods, pp. 222–229. Springer, Heidelberg (2003)
86. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Real-time optic flow computation with variational methods. *IEEE Trans. of Image Processing* 14(5), 608–615 (2005)
87. 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)*, London, UK (September 2009)
88. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences, pp. 1–6 (2008)
89. Meister, S., Jähne, B., Kondermann, D.: Outdoor stereo camera system for the generation of real-world benchmark data sets. *Optical Engineering* 51 (2012)
90. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: *Computer Vision and Pattern Recognition (CVPR)*, Providence, USA (June 2012)
91. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8 (2008)
92. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision* 77(1), 157–173 (2008)
93. Meister, S.: *A study on ground truth generation for optical flow*. Master's thesis, University of Heidelberg (2010)
94. Meister, S., Kondermann, D.: Real versus realistically rendered scenes for optical flow evaluation. In: *Proceedings of 14th ITG Conference on Electronic Media Technology* (2011)
95. Haeusler, R., Klette, R.: Benchmarking Stereo Data (Not the Matching Algorithms). In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) *DAGM 2010*. LNCS, vol. 6376, pp. 383–392. Springer, Heidelberg (2010)
96. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, vol. 2, p. 3 (2011)
97. Kaneva, B., Torralba, A., Freeman, W.: Evaluation of image features using a photo-realistic virtual world. In: *Proc. of the 13th International Conference of Computer Vision, ICCV 2011* (2011)
98. Beizer, B.: *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc. (1995)

# Camera-Based Fall Detection on Real World Data

Glen Debard<sup>1,6</sup>, Peter Karsmakers<sup>1,5</sup>, Mieke Deschodt<sup>2,4</sup>, Ellen Vlaeyen<sup>2,4</sup>,  
Eddy Dejaeger<sup>4</sup>, Koen Milisen<sup>2,4</sup>, Toon Goedemé<sup>3,6</sup>, Bart Vanrumste<sup>1,5,7</sup>,  
and Tinne Tuytelaars<sup>6,7</sup>

<sup>1</sup> MOBILAB: Biosciences and Technology Department, KHKempen, Belgium

<sup>2</sup> Center for Health Services and Nursing Research, KU Leuven, Belgium

<sup>3</sup> Lessius Mechelen, Campus De Nayer, Belgium

<sup>4</sup> Department of Internal Medicine, Division of Geriatric Medicine, University  
Hospitals Leuven, Belgium

<sup>5</sup> ESAT-SCD, KU Leuven, Belgium

<sup>6</sup> ESAT-PSI, KU Leuven, Belgium

<sup>7</sup> IBBT Future Health Department, Belgium

**Abstract.** Several new algorithms for camera-based fall detection have been proposed in the literature recently, with the aim to monitor older people at home so nurses or family members can be warned in case of a fall incident. However, these algorithms are evaluated almost exclusively on data captured in controlled environments, under optimal conditions (simple scenes, perfect illumination and setup of cameras), and with falls simulated by actors.

In contrast, we collected a dataset based on real life data, recorded at the place of residence of four older persons over several months. We showed that this poses a significantly harder challenge than the datasets used earlier. The image quality is typically low. Falls are rare and vary a lot both in speed and nature. We investigated the variation in environment parameters and context during the fall incidents. We found that various complicating factors, such as moving furniture or the use of walking aids, are very common yet almost unaddressed in the literature. Under such circumstances and given the large variability of the data in combination with the limited number of examples available to train the system, we posit that simple yet robust methods incorporating, where available, domain knowledge (e.g. the fact that the background is static or that a fall usually involves a downward motion) seem to be most promising. Based on these observations, we propose a new fall detection system. It is based on background subtraction and simple measures extracted from the dominant foreground object such as aspect ratio, fall angle and head speed. We discuss the results obtained, with special emphasis on particular difficulties encountered under real world circumstances.

**Keywords:** Fall Detection, Video Surveillance, Assisted Living.

## 1 Introduction

Many older persons fall and are not able to get up again unaided. Thirty to forty-five percent of the persons aged 65 or older living at home and more than half of the elders living in a nursing home fall at least once a year. One out of three up to one out of two older persons fall more than once every year [14,24].

Ten to fifteen percent of those who fall, suffer severe injuries. [14] The lack of timely aid can lead to further complications such as dehydration, pressure ulcers and even death. Although not all falls lead to physical injuries such as hip fracture, psychological consequences are equally important, leading to fear of falling, losing self-confidence and fear of losing independence [4,14]. Taking the ongoing aging of the population into account, it is obvious that adequately detecting fall incidents is getting more and more important. Indeed, a large study in the Netherlands reported an increase of fall-related hospital admissions from 1981 to 2008 by 137% [8]. Furthermore, falls are associated with substantial costs. For instance, the excess costs associated with treating hip fractures range between USD 11,241-18,727 in the first year following the fracture [7]. A study in the U.K. estimated the total cost (year 1999) related to injurious falls in those aged 75 and older to be almost 647 million [20].

The existing technological detectors are mostly based on wearable sensors. However, a market study of SeniorWatch [21] discovered that the sensors are not worn at all times (e.g. at night). Also, in case the device is button operated, as with a Personal Alarm System, some persons with (mild) cognitive impairment are not always able to activate the alarm system due to complexity of issues around the use of call alarms [4]. As a result, many falls remain undetected. A camera-based system, on the other hand, has the potential to overcome the limitations mentioned above, because it is contactless and does not require initiative of the person. On the downside, one or more cameras need to be installed in every room, increasing the cost of this system; the system is fixed; and only works indoor. Another disadvantage is that it is not possible to take the system along on a trip.

In the last decade, several research groups have focused on a camera based fall detection algorithm. However a major drawback of these studies, is the fact that they use simulated data. The falls have been recorded in artificial environments and the simulators are mostly younger persons. The goal of our work is the development and evaluation of a prototype camera based fall detection system using real life data. For this, we have installed cameras monitoring four older persons with an increased risk of falling at their place of residence for six months. Three of these persons are residing in a nursing home, since people with a history of falling are often institutionalized.

In the remainder of this paper, we first discuss how we captured our dataset and the challenges posed by the usage of real world data (Section 2). Next, we give an overview of earlier work (Section 3). In Section 4, we describe the fall detection algorithm we developed, followed by some preliminary results of the validation of our algorithm using the real life video data in Section 5. In Section 6 we discuss these results. Section 7 concludes the paper.

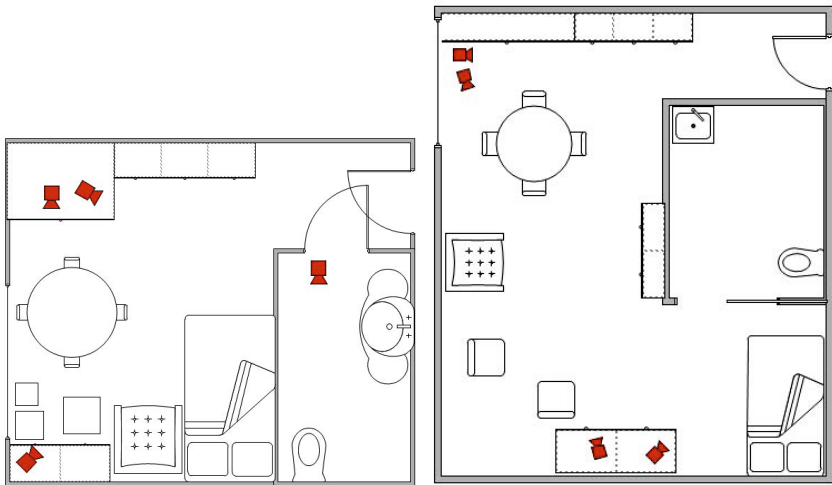
## 2 The Dataset and Its Challenges

### 2.1 Data Collection

During the acquisition phase, we have installed four camera systems at the place of residence of four older persons. one at the home of an independently living older woman, one in a room of a nursing home and two in a service flat. Figure 1 shows how the cameras were installed in the nursing home. For privacy reasons, we did not plan to install a camera in the bathroom. However, the person in the nursing room asked us to install a camera there after falling twice at that location. We also provided a control panel that allowed the participant to switch off the system whenever wanted. However, only the cleaning personnel used this option.

The participants' age was in the range of 83 to 95 years old, and all of them had an increased risk of falling. Recordings were made during approximately six months, 7 days a week, and 24 hours a day. During these six months, we recorded over 14.000 hours of video and captured 24 falls. Most falls occurred in two persons. The person living independently did not fall during our monitoring period, while one of the participants in a service flat only fell once. To our knowledge, this is a unique dataset. To capture these events, we received the approval of the Medical Ethics Committee of the Leuven University Hospitals and all participants gave their written informed consent.

For each residence we used 4 wall-mounted IP cameras. We used a combination of ACTI ACM-1511 and AXIS 207 cameras. The ACTI cameras already had day/night vision. We changed the lenses of the AXIS cameras to one with a view angle of 80 degrees without a near-infrared filter. Additional near-infrared sources made it possible to record video in low-light conditions and during the



**Fig. 1.** Setup of cameras. Left panel: Room in nursing home. Right panel: Service flat.



night. We recorded images with a resolution of 640 by 480 pixels using a frame rate of 12 frames per second. Since we wanted to be able to analyze images in low light conditions or during the night, we used gray level images. To be able to store the data, we used M-JPEG compression. This reduced the disc space usage to 1.8 GB per hour.

Not only did the collection of this dataset allow us to evaluate prototype systems for camera based fall detection on real world data (see Section 5), it also provided us with valuable insights on the typical challenges that can arise when using real life data, both for fall and non-fall scenarios. While we cannot make the dataset publicly available because of privacy issues, we can comment on these general findings.

## 2.2 Data Characteristics in a Typical Real Life Scenario

The analysis of the captured video shows some challenges that researchers developing fall detection systems should be aware of. Which ones are important depends on the algorithms used.

*Image quality.* First, the quality of the camera in a real world scenario is typically lower than what is used in a lab setup. Indeed, from a practical point of view, to be cost-efficient, it is not possible to install high quality cameras. Moreover, it is necessary to monitor the person also in low-light conditions during the evening or night. Therefore, we also needed to record near-infrared, which is often more noisy. It is important to install as few cameras as possible. The usage of a camera and lens with view angle close to 90 degrees installed in the corner in of the room gives the best coverage. But the wide angle of the lens also decreases the spatial resolution of the camera.

*Color information.* In near-infrared night images, no color information is available. But even during daytime when color information is available, it is not very reliable. Especially the different light sources in a house (sun light, fluorescent light, light bulbs, tv-screen, etc.) present some specific challenges. For example, during one of our preliminary tests, a person moved in front of a window, the sunlight was partially blocked, which changed the color of the incident light. Several methods for fall detection proposed in the literature [2] rely on color-based shadow detection algorithms to improve the output of a background subtraction algorithm. However, these are based on the assumption that when an area is covered by a shadow, this results in a significant change in brightness only without change in color information [6]. This assumption is not always met in real world circumstances. Hence color can be an unreliable source of information.

*Overexposure.* The range of light intensities that occur during the day, is extensive. A good configuration of the camera is needed. Even then, the brightness of the sun can cause overexposure in some areas of the image. Careful placement of the cameras in the room can decrease the problem to some extent. Instead of pointing the camera to the window, it is better to attach it above the window,



**Fig. 2.** Examples of video frames with different illumination. Upper left: Sunlight causes overexposure at window. Upper right: Localized overexposure caused by halogen lamps. Lower left: Same room with minor overexposure. Lower right: Frame recorded at night using near infrared.

facing the room. However, since it is necessary to cover all areas of the room with a limited number of cameras, pointing them towards the windows cannot always be avoided. Also halogen lamps can cause overexposure, as well as special lighting conditions. Figure 2 shows an example of the same room at different moments with different kinds of illumination.

*Image clutter.* Not only the change in illumination has to be taken into account, but also the changes occurring in the room itself. Rooms are often small, both in nursing homes as in private homes and older persons tend to collect a lot of furniture, which can have a sentimental value. When moving to a smaller residence, they want to take these along. As a consequence, rooms are often highly cluttered. When moving around in the room, the person is quite often partially occluded. Over longer time periods, furniture is also less static than one might expect (see also Figure 2). Furniture that is shifted, should therefore be dealt with appropriately by the system.

*Walking aids.* Some older persons have difficulties walking unaided. Because of this, they sometimes use a walking aid like e.g. a rollator or a walking frame. The legs of the person and part of the lower body can be occluded by this. Moreover,

the walking aid is another dominant foreground object, sometimes moving along with the person, sometimes put aside (see e.g. Figure 2 top left). Fall detection algorithms that rely on the person being the only or largest foreground object in the scene may not be able to cope with this situation.

*Appearance changes.* The appearance of the person also changes over time, e.g. while getting (un)dressed or changing clothes. Under such conditions, relying on color or intensity distributions to track the person, may not be a good idea.

*Other moving objects.* Other challenges are for example a television or a cupboard with doors that can be opened. Also a door is difficult to take into account. It is a large moving object, and what is behind the door can differ each time (e.g. an entrance door in a nursing home). A person that is lying in bed, is almost completely occluded by the sheets while sleeping. But getting out of bed, the sheets are folded back, which again represents a large moving object. Some methods based on motion history images (e.g. [19]) learn to ignore the motion in these image areas. However, this means that falls occurring at these locations are more likely to be ignored as well.

*Motion patterns.* The behavior of an older person can differ significantly from that of a younger person. Analyzing our data, we observed that some persons stay seated in the same place for extended times during the complete day. The manner in which older persons move can differ significantly from younger persons, certainly with respect to the speed of movement, which can be extremely slow in some cases. Also the typical gait is different, with shorter strides.

### 2.3 Analysis of the Observed Fall Incidents

As mentioned before, we monitored four persons and collected 24 falls. One person did not fall during the monitoring period, while a second person fell only once. The other two persons fell 10 and 13 times, respectively. Because the majority of the falls occurred in only two individuals, it is not possible to generalize our findings. Nevertheless, the recorded falls already give us some insight in the challenges their detection represents.

*Use of walking aids.* Both persons with a high number of falls, often used a rollator walker. Half of the falls ( $n=12$ ) occurred while using a walking aid. When the person was falling, the rollator was pushed forward, sometimes crossing a huge part of the room, or turning over. All these cases may interfere with the fall detection, either because the person is occluded behind it, or because it corrupts the extracted features. Figure 3 shows some examples of interference that a rollator walker can cause.

*Initial pose.* Not all falls start from a standing pose. A fall can also start from a crouching or a bend over position, while picking something up. This occurs in five falls (21%). A fall can also happen in two steps. Sometimes the person was



**Fig. 3.** Two fall incidents with interference of a rollator walker. Upper panels: a fall where the rollator partially occludes the person. Lower panels: the rollator is pushed and rolls away from the person.

able to grab hold of a door or chair, but after a short time, had to let go and fell to the ground. This happened in two falls (9%). Five falls (21%) happened shortly after standing up or while preparing to sit down. This arises because an older person sometimes doesn't have enough strength in his/her legs to stand up or sit down slowly.

*Occlusions and appearance.* Occlusions are another important challenge. In eleven falls (46%), the person was completely or partially occluded, either by the walking aid or by the furniture. In one case, the fall started in one room and ended in an adjacent one. Even with multiple cameras in the room, it is often impossible to get an unoccluded view of the person. In three falls (12%), the person was undressing, which drastically changed the appearance of the person.

*Other moving objects.* One of the most occurring challenges are other moving objects in the scene. In 18 falls (75%), the furniture in the room was moved by the fall. Certainly chairs and tables are shifted easily, but also small and even larger cupboards can be moved during a fall. Moving doors are also common. In one case, a painting on the wall was shifted. The consequence is that sometimes the appearance of the room can change completely. We already mentioned that in some cases, the room is really filled with different pieces of furniture. In such a



**Fig. 4.** Two fall incidents with moving furniture. Upper panels: The table and chairs are moved and the upper body of the person is occluded. Lower panels: The table, chairs and sofa are moved. The rollator is also fallen over and the person is almost completely occluded.

case, it is almost impossible to not hit something while falling down. Even when a room is only modestly furnished, a fall against furniture will occur in most cases. Figure 4 shows some examples of this type of interference. Especially methods assuming a static scene and relying on background subtraction are affected by this. On the other hand, a sudden motion over a large part of the scene could by itself be a cue for fall detection.

*Unbalanced data.* The final challenge is the ratio of fall to non-fall data. We have recorded a dataset that is really extensive. The persons that we monitored all had a high risk of falling. The numerous falls of two of our participants show this. But even in this case, the falls only represent a tiny portion of the available data. The performance of a fall detector is not only determined by its ability to detect a fall, but also by its ability to generate as few false alarms as possible. To test this, it is important to not only use the falls, but also part of the realistic non-fall data.

The usage of this real life data and the numerous challenges it represents, greatly increases the complexity in building a working fall detection system. In the following section we review the state-of-the-art, taking the challenges mentioned above into account. Next, in Section 4, we explain our preliminary fall detector in more detail.

### 3 Related Work

Most systems described in the literature can be divided in two main approaches to the problem: those that try to detect the action of falling directly (e.g. [1,2,5,12,13,18,19,22,23,25]), and those that instead detect unusual events in general (e.g. [15,16,19]). The latter rely on indirect evidence, such as prolonged inactivity at unusual locations, to infer fall incidents. Since normal behavior in terms of person appearance (actions or poses) is considered too broad and varied to model, these systems typically focus on spatio-temporal trajectories instead. By doing so, the problem of the large variability in appearance is circumvented. Moreover, since it is only needed to learn what normal behavior looks like, the unbalancedness of the data is not really an issue, nor is the variability in appearance of fall incidents. On the downside, what is normal behaviour in terms of spatio-temporal trajectories is typically location and camera (viewpoint) specific. Therefore, these systems usually need to be retrained for each new camera setup. Also, an unusual pattern does not imply the occurrence of a fall incident (or another event that would require intervention, for that matter). If, for instance, a person is ill, he/she may show various forms of unusual behaviour, such as staying in bed longer than usual, or going to the bathroom in the middle of the night. This may result in lots of false alarms.

Methods that more directly try to detect the dynamic event of falling, do not suffer from the above mentioned limitations. In this category, we again distinguish between methods building on simple cues like motion detection, often combined with domain knowledge (e.g. [1,2,13,18,19,23]), and methods that build on recent advances in generic person detection and action recognition (e.g. [22]). While the latter may seem promising at first, the amount of training data seems insufficient to learn a reliable model for falls, especially when taking the large variability in appearance of the falls into account. Also the quality of the images is a limiting factor. Figure 5 shows the output of a state-of-the-art person detector / pose estimator [28] applied to some of our recordings. A tracker might improve these results to some extent, but we doubt whether it will be accurate enough to infer a fall from the change in pose. Finally, the needed computation time of these methods often does not allow for real-time processing.

It is possible to use more complex methods, like action recognition and person detection, but we believe the most promising approach at this moment to be a combination of relatively simple, low-level cues with available domain knowledge. Since we know the cameras are static, background subtraction can be applied to find the moving foreground objects, including the person. Likewise, one can build on domain knowledge to design simple yet robust fall features, such as the aspect ratio [1,2,13,27] or the speed of the head of a person [18,12] (exploiting the fact that the head remains mostly unoccluded). These can be combined in a low dimensional representation and presented to a classifier, with limited risk of overfitting. Background subtraction has been used by many systems (e.g. [1,2,5,12,13,16,23,27]). However, in many cases, it is assumed that this results in an accurate silhouette of the person, based on which the pose can be determined (e.g. [1,2,5,16]). This is usually not the case for our real life data.



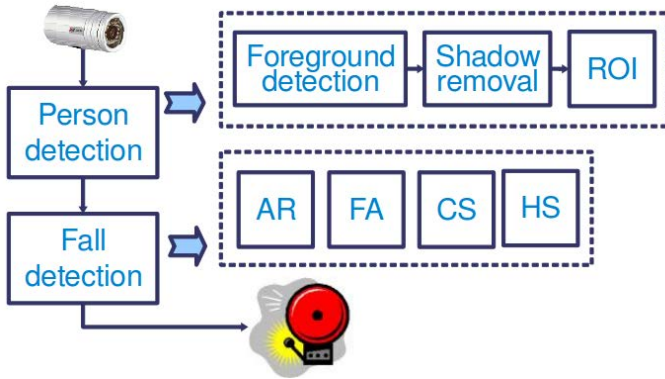
**Fig. 5.** Examples of the output of a state-of-the-art person detector [28]. Upper left panel: Successful detection of human pose. Upper right and lower panels: Failed to detect pose of the person. (Green : head, yellow : torso, violet : left arm, light blue : right arm, red : left leg, dark blue : right leg)

Due to the low image quality as well as problems with overexposure, occlusions or changing illumination conditions, background subtraction (even after shadow removal) only gives a rough idea of where the person might be. Also the fact that older persons often stay seated at the same place over long periods of time does not help in this respect.

In conclusion, methods exploiting relatively low-level cues (e.g. [10,18,19,23]) seem most promising in a real life context. They are robust, fast to compute, and relatively generic (no need for retraining or calibration for each new camera setup). More complex schemes can then be added as further verification or to corroborate the results, if applicable.

## 4 Methods

Our fall detection algorithm consists of four main parts: video acquisition, person detection, fall detection and alarm generation (see Figure 6). The video is first converted to gray level images. This way there is no need to alter the processing if we switch to near-infrared at night. The alarm generation is not implemented at this stage. The next sections explain the person detection, features for fall detection and fall detector in further detail.



**Fig. 6.** Overview of the system (ROI: region of interest detection; Different fall features: Aspect Ratio (AR), Fall Angle (FA), Speed of center of gravity (CS), Head Speed (HS))

#### 4.1 Person Detection

**Foreground Detection.** We first needed to segment out the foreground. For this we used a background subtraction technique based on an approximate median filter [11]. The advantages of the approximate median filter are its low memory consumption, fast computation and robustness. The drawbacks are its rather slow update to large changes in illumination and the fact that, as any background subtraction method using a dynamic background, the foreground is influencing the background. This influence leads to the appearance of a ghost figure. When a person is sitting on the couch for a longer period, the background is updated to incorporate the person into the background. If he stands up, the region of the couch that was occluded previously will also differ from the background and it is detected as foreground. This can influence the extraction of the features to detect a fall. Not updating the model within the detected ROI (see below) is not a solution, since a background model that is not updated over a longer time is also not representative anymore due to changes in lighting conditions.

**Shadow Removal.** A shadow cast by a moving object is also detected as foreground since it makes the covered pixels appear darker. This makes the detected foreground region larger than it should be. To remove this shadow, we used the property that a shadow only changes the intensity of the pixel while the texture of the covered region does not change [6]. As a result, the texture of the shadow is correlated with the corresponding texture of the background image. Jacques and Jung describe in [9] the usage of the cross correlation (CC) to see how good the detected foreground pixels match the background pixels. In case the cross correlation is higher than a certain threshold and the pixel is darker in the current image, then the pixel is classified as shadow. Also other changes in illumination can be eliminated using this technique when removing





**Fig. 7.** Extraction of fall features: purple: bounding box, white: bounding ellipse, green: center of gravity, blue: head position (The black box is for privacy reasons)

the constraint that the pixel has to be darker in the current image. Jacques and Jung state that a threshold for the cross correlation of 0.98 together with a  $5 \times 5$  neighborhood gives a good result. These values were also used in our experiments.

**ROI Detection.** The next step in our algorithm was the determination of a region of interest (ROI). We first used an erosion/dilation step on all foreground pixels. Next, we applied a connected components analysis to determine the foreground objects. The largest object in the foreground was selected and considered to correspond to the person. As noted earlier, selecting the largest foreground object is prone to errors, since furniture or walking aids may move as well. A better choice is to rely on a tracker. However, this was left as future work. To minimize noise and interference, the object had to be larger than a certain threshold. In our case, a minimum of 17500 pixels gave the best performance. From this object we started to extract the features to detect a fall.

## 4.2 Fall Detection Features

Using the person, we extracted four features to detect a fall, including: aspect ratio (AR) [12][13][27], fall angle (FA) [19][27], center speed (CS) [19] and head speed (HS) [5][12] (see Figure 7). These features have been designed based on domain knowledge, i.e. in such a way that they capture relevant information to discriminate falls from other actions, while at the same time being robust to inaccuracies in the person detection. These are also the most widely used in the literature, as explained in Section 3.

**Aspect Ratio.** The aspect ratio is calculated as the ratio of the width of the bounding box (BB) around the foreground object and its height. A low aspect

ratio represents an upright person, while a high aspect ratio might point to a person lying down.

**Fall Angle.** The angle of the person in the image can be defined as the angle between the long axis of the bounding ellipse and the horizontal direction. A person that is standing, has an angle close to 90 degrees. A small angle represents a person lying down (if seen from a side-view). We defined the fall angle as the change in angle over a fixed timespan (2 seconds in our experiments). A large fall angle can indicate a fall.

**Center Speed and Head Speed.** A person, and certainly an older person, typically moves with a low speed. In contrast, most of the falls have a portion with high speed movement. Based on this observation, we used two fall features related to speed, center speed and head speed. Center speed is the speed of the center of gravity of the foreground object. This center of gravity has the advantage that it is rather stable. Small changes in appearance of the person give only small changes in the center of gravity. But an occlusion of the lower body, which happens frequently, causes the center of gravity to move upwards. The head, on the other hand, is visible in most non-fall actions. In [5] Foroughi *et al.* define the head as the highest point of the object. Here we used the highest end of the main axis of the bounding ellipse as head position. The speed itself was then defined as the amount of pixels that the point had shifted between two adjacent frames in the video divided by the time between these two frames.

### 4.3 Fall Detection with SVM

Given that the features defined in the previous section are based on domain knowledge, each of them can be used as a basic fall detector simply by choosing an appropriate threshold (as done e.g. in [27]). However, better results can be obtained if they are merged, and a single classifier combining the different cues is learned. In this section we propose a Support Vector Machine (SVM) [26] based fall detector which classifies a time slot (by its features) either as a fall or as another event.

As noted earlier, the classes are imbalanced (in most cases "normal" behavior is seen, falls are rare) and class distributions are overlapping (the limited set of features being used might not clearly discriminate all "normal" events from falls). Without any precautions SVM prediction might result in a simple majority vote ignoring the existence of falls. To address this problem the SVM learning objective was modified such that different weights are applied to misclassifications depending on the class [17]. In the SVM learning objective errors for the minority class were multiplied by  $w$  while majority errors were multiplied by  $1 - w$ . How we determined  $w$ , is explained later.

In order to validate the fall detector the available dataset was randomly partitioned into a training set, containing 66% of the data, and an independent test set with the remaining data. The training set was then used to estimate the

SVM model parameters and a set of hyper-parameters. The test set was only used for evaluation.

The hyper-parameters used in this paper are (a) the weight  $w$ , (b) the regularization parameter of the SVM and (c) the Radial Basis Function (RBF) kernel bandwidth. These were selected using cross-validation and a grid search maximizing the Area Under the Curve (AUC) of a Receiver Operating Characteristic (ROC) curve. The ROC curve was computed by varying the threshold on the distances of considered data examples to the separating hyperplane which is defined by the SVM model. In order to reduce random effects induced by partitioning the data averaged AUC scores were computed on different data partitionings.

Additionally, feature selection was performed by executing a greedy forward search. Firstly, 4 univariate SVM models (each based on 1 different feature and trained using the procedure explained above) were compared in terms of AUC. Next, the best feature (corresponding to the best SVM model) was retained and combined with each of the remaining features in a bivariate SVM model. The best feature set was retained and the procedure was repeated to find the best feature set with incremented cardinality. Note that features were standardized to have zero mean and unit standard deviation.

## 5 Results

As mentioned before, we acquired an extensive dataset. To validate the algorithm, we used for each of the 24 falls, the camera on which the person is best visible. From this video, we selected a fragment of 20 minutes with the fall occurring in the last two minutes of the video. Our current system does not use the post-fall information (i.e., the person lying on the floor). Each video was divided in non-overlapping time slots of two minutes long. For each time slot, the fall features were extracted and the maximum values during that time slot were used for further analysis (max pooling). In total this resulted in 240 epochs, of which 24 are labeled as a fall. In a real system, the choice of the cameras could be dealt with using a voting mechanism. The extraction of the different fall features was executed on a pc with an Intel Core2 Quad Core Q9650 CPU running at 3 GHz. The algorithm was implemented in C++ using OpenCV. We can run four threads with different video, each processing around eight frames per second.

Given our four features, SVM models were estimated using the procedure described in the previous section. Results were averaged over 10 different partitionings of training and test set.<sup>1</sup> Table 1 lists the averaged AUC scores and the corresponding standard deviations for SVM models based on different feature sets. Figure 8 and Figure 9 respectively present the ROC and Precision Recall curves of the four best performing SVM models (measured in terms of AUC). It can be observed that the combination of aspect ratio and head speed is to be preferred. Using this feature set SVM outputs an averaged operating point with a recall of  $0.9(\pm 0.2)$  and a precision of  $0.26(\pm 0.07)$ . Another observation is that the fall angle performs significantly lower than the other features.

<sup>1</sup> Note that for each feature set the same set of data partitionings was used.

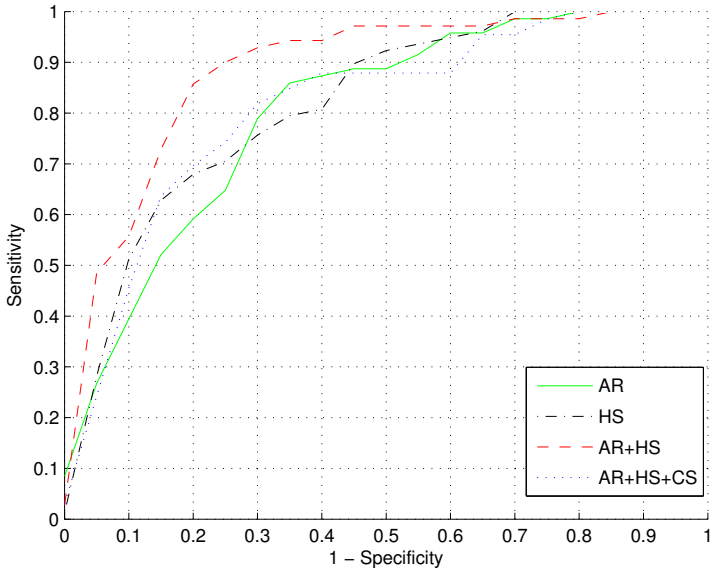


Fig. 8. ROC Curve

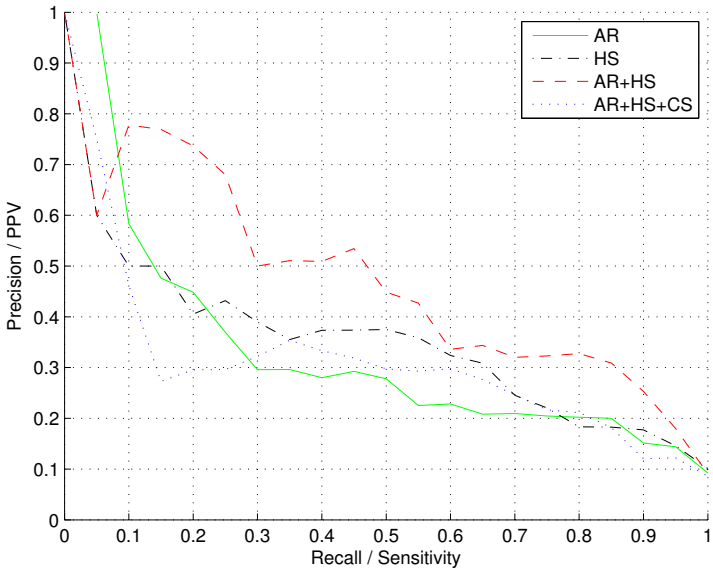
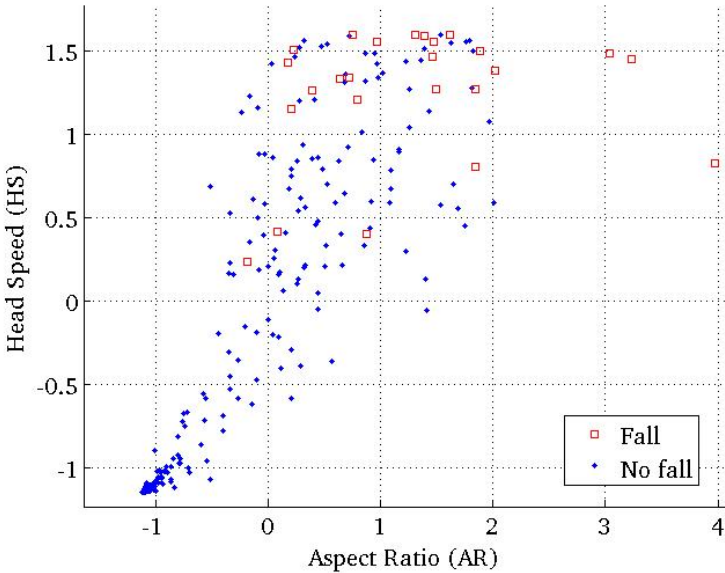


Fig. 9. Precision Recall graph

**Table 1.** Fall detection results

Feature set	AUC
{AR}	0.88(±0.06)
{FA}	0.53(±0.09)
{CS}	0.84(±0.05)
{HS}	0.87(±0.05)
{AR, HS}	0.91(±0.06)
{AR, HS, FA}	0.90(±0.05)
{AR, HS, FA, CS}	0.86(±0.06)



**Fig. 10.** Class distribution using normalized aspect ratio and head speed

Considering Figure 9, we noticed that the precision quickly drops when increasing the recall. This behavior can be explained by looking at Figure 10, that represents the distribution of the data when considering features aspect ratio and head speed. Here we can see that there are quite a number of non-falls that are close to the falls. Closer visual inspection revealed that 90% of these have 4 main causes. In 25% of the cases 2 persons were present in the room. In 20% of the cases another foreground object had almost the same size as the person. In both cases, the system often switched to the other person or object, resulting in large motions and changes in aspect ratio. In 25% of the cases, the person’s image was split in 2 blobs which were almost the same size. Situations where such an event occurs included: over-illumination, the person wearing a shirt that is similar to the background or the person starting to be integrated in the background by the background update. This often resulted in a deviating aspect ratio as well as large motions as the

system jumps back and forth between the different parts. Finally, in 20% of the cases there was interference of a ghost figure or moved furniture.

## 6 Discussion

Comparing our results with those reported in the literature [13, 18], we have a similar or higher detection rate, but a higher false alarm rate. Two out of the three undetected falls started and ended outside of the view of the camera. This was e.g. the case when the older person was taking something out of the closet. The door was occluding the person at the start of the fall. During the fall the person was visible just for a very short time, before tumbling in the bathroom. A better placing or additional cameras can solve this. The higher false alarm rate can be explained by the challenging nature of our dataset, including various sources of errors that were previously largely ignored. In real life, falls only occur in rare cases. It is thus important to significantly decrease the number of false alarms to an acceptable level to get a usable fall detection system.

Most of the false alarms can be solved by using more advanced techniques. The largest improvement can be expected from the use of a tracker. This avoids large motions and changes in appearance caused by jumping back and forth between different foreground blobs of different (parts of) persons or other objects. This is the first step that we will investigate further. Also a more advanced foreground detection, that is robust to continuous changes in illumination, slow movement of older persons, different types of light-sources and possible over-illumination, can give a large improvement. Using a mixture of Gaussians to model the background showed no improvement on first sight. A means to detect a person in the foreground, like for example the person detector of Felzenszwalb *et al.* [3] can also reduce erroneous foreground objects. This detector is only trained for standing persons (both whole body and upper body), but it can still help as a verification every now and then. Alternatively, an articulated pose estimator such as [28] may be used as well. In Figure 5 it did not perform well. However, given a good initialization based on foreground detection, it may be useful.

Additional improvements may be possible by adding other fall features (e.g. posture or other appearance-based approaches), integrating information of several cameras or other sensors and especially by integrating the post-fall information.

In our tests, we used the camera on which the person was best visible. In a real system, this choice has to be made automatically. A voting mechanism, that uses the information how certain the system is that a fall occurred, can be implemented for this. This knowledge of the certainty of the fall can also be used to determine the needed action.

To reduce the annoyance of the false alarms, it is also possible to use an alarming chain. A possible fall could first be presented to the resident itself, if he doesn't react, a further escalation to different levels of caregivers can be executed.

## 7 Conclusion

Fall detection is becoming more and more important to ease the fears of an older person or someone with an increased fall risk. In this way these persons are able to live longer independently in a more comfortable way. In this paper we have given an overview of our ongoing research, which is unique in the way we use real life data. We have shown that under real life conditions, various sources of errors emerge such as other persons, moving furniture, walking aids, etc. that significantly increase the number of false alarms, yet have previously been largely ignored. Our preliminary fall detector shows a recall of  $0.9(\pm 0.2)$  and a precision of  $0.26(\pm 0.07)$ . This calls for further research into more discriminative fall features, as well as better foreground detection algorithms, including tracking and person detection.

**Acknowledgments.** This work is funded by the FWO via project G039811N: "Monitoring van gedrag en ongebruikelijke menselijke activiteit met meerdere camera's", by the IWT via TETRA project 80150 "Fallcam: Detection of fall in older persons with a camera system." and by the EU via ERASME (FP7) project IWT 100404 "AMACS: Automatic Monitoring of Activities using Contactless Sensors." The authors like to thank the persons who participated in the research by giving their permission to be monitored during several months.

## References

1. Anderson, D., Keller, J., Skubic, M., Chen, X., He, Z.: Recognizing falls from silhouettes. In: 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2006, pp. 6388–6391 (September 2006)
2. Cucchiara, R., Prati, A., Vezzani, R.: An intelligent surveillance system for dangerous situation detection in home environments. *Intelligenza Artificiale* 1(1), 11–15 (2004)
3. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Anchorage, Alaska (June 2008)
4. Fleming, J., Brayne, C.: Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. *British Medicine Journal* 337(v17 1), 2227 (2008)
5. Foroughi, H., Aski, B., Pourreza, H.: Intelligent video surveillance for monitoring fall detection of elderly in home environments. In: 11th International Conference on Computer and Information Technology, ICCIT 2008, pp. 219–224 (2008)
6. Grest, D., Frahm, J., Koch, R.: A color similarity measure for robust shadow removal in real-time. *Vision, Modeling and Visualization* (2003)
7. Haentjens, P., Lamraski, G., Boonen, S.: Costs and consequences of hip fracture occurrence in old age: An economic perspective. *Disability and Rehabilitation* 27(18-19), 1129–1141 (2005)
8. Hartholt, K.A., van der Velde, N., Looman, C.W.N., van Lieshout, E.M.M., Pannekoek, M.J.M., van Beeck, E.F., Patka, P., van der Cammen, T.J.M.: Trends in fall-related hospital admissions in older persons in the Netherlands. *Arch. Intern. Med.* 170(10), 905–911 (2010)

9. Jacques, J.C.S., Jung, C.R.: Background subtraction and shadow detection in grayscale video sequences. In: *The XVIII Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI 2005* (2005)
10. Lee, T., Mihailidis, A.: An intelligent emergency response system: preliminary development and testing of automated fall detection. *Journal of Telemedicine and Telecare* 11(4), 194–198 (2005)
11. McFarlane, N.J.B., Schofield, C.P.: Segmentation and tracking of piglets in images. *Machine Vision and Applications* 8(3), 187–193 (1995)
12. Miao, Y., Naqvi, S., Chambers, J.: Fall detection in the elderly by head tracking. In: *IEEE/SP 15th Workshop on Statistical Signal Processing, SSP 2009*, pp. 357–360 (September 2009)
13. Miaou, S.G., Sung, P.H., Huang, C.Y.: A customized human fall detection system using omni-camera images and personal information. *Distributed Diagnosis and Home Healthcare*, 39–42 (2006)
14. Milisen, K., Detroch, E., Bellens, K., Braes, T., Dierickx, K., Smeulders, W., Teughels, S., Dejaeger, E., Boonen, S., Pelemans, W.: Falls among community-dwelling elderly: a pilot study of prevalence, circumstances and consequences in flanders. *Tijdschr Gerontol Geriatr* 35(1), 15–20 (2004)
15. Nait-Charif, H., McKenna, S.J.: Activity summarisation and fall detection in a supportive home environment. In: *ICPR 2004: 17th International Conference on Proceedings of the Pattern Recognition*, vol. 4, pp. 323–326. IEEE Computer Society, Washington, DC (2004)
16. Nater, F., Grabner, H., Van Gool, L.: Visual abnormal event detection for prolonged independent living. In: *International Mobile Health (mHealth) Workshop* (2010)
17. Osuna, E., Freund, R., Girosi, F.: Support vector machines: Training and applications. *AI Memo 1602*, Massachusetts Institute of Technology (1997)
18. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Monocular 3d head tracking to detect falls of elderly people. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2006)
19. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Fall detection from human shape and motion history using video surveillance. In: *21st International Conference on Advanced Information Networking and Applications Workshops, AINAW 2007*, vol. 2, pp. 875–880 (2007)
20. Scuffham, P., Chaplin, S., Legood, R.: Incidence and costs of unintentional falls in older people in the United Kingdom. *J. Epidemiol. Community Health* 57(9), 740–744 (2003)
21. SeniorWatch: Fall detector: Case study of european ist seniorwatch project. Tech. rep., SeniorWatch (2001)
22. Syngelakis, E., Collomosse, J.: A bag of features approach to ambient fall detection for domestic elder-care. In: *Proc. Intl. Symp. on Ambient Technologies, AMBIENT 2011* (2011)
23. Thome, N., Miguet, S., Ambellouis, S.: A real-time, multiview fall detection system: A lhmm-based approach. *IEEE Transactions on Circuits and Systems for Video Technology* 18(11), 1522–1532 (2008)
24. Tinetti, M.E.: Preventing falls in elderly persons. *New England Journal of Medicine* 348(1), 42–49 (2003)



25. Töreyn, B.U., Dedeoğlu, Y., Çetin, A.E.: HMM Based Falling Person Detection Using Both Audio and Video. In: Sebe, N., Lew, M., Huang, T.S. (eds.) HCI/ICCV 2005. LNCS, vol. 3766, pp. 211–220. Springer, Heidelberg (2005)
26. Vapnik, V.: Statistical learning theory. Wiley, New York (1998)
27. Willems, J., Debar, G., Vanrumste, B., Goedemé, T.: A video-based algorithm for elderly fall detection. In: Medical Physics and Biomedical Engineering World Congress, WC 2009 (2009)
28. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1385–1392 (June 2011)

# Semantic Structure from Motion: A Novel Framework for Joint Object Recognition and 3D Reconstruction

Sid Yingze Bao and Silvio Savarese

The University of Michigan at Ann Arbor, MI, USA

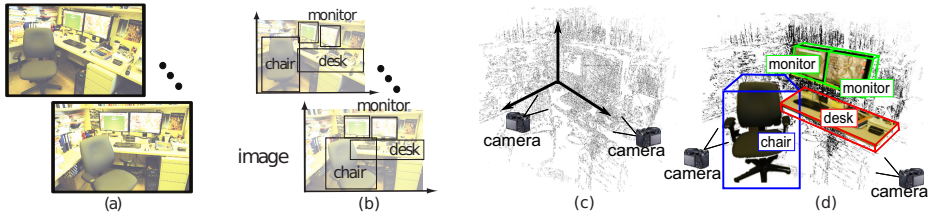
{yingze,silvio}@eecs.umich.edu

<http://www.eecs.umich.edu/vision/projects/ssfm/index.html>

**Abstract.** Conventional rigid *structure from motion* (SFM) addresses the problem of recovering the camera parameters (motion) and the 3D locations (structure) of scene points, given observed 2D image feature points. In this chapter, we propose a new formulation called *Semantic Structure From Motion* (SSFM). In addition to the geometrical constraints provided by SFM, SSFM takes advantage of both semantic and geometrical properties associated with objects in a scene. These properties allow to jointly estimate the structure of the scene, the camera parameters as well as the 3D locations, poses, and categories of objects in a scene. We cast this problem as a max-likelihood problem where geometry (cameras, points, objects) and semantic information (object classes) are simultaneously estimated. The key intuition is that, in addition to image features, the measurements of objects across views provide additional geometrical constraints that relate cameras and scene parameters. These constraints make the geometry estimation process more robust and, in turn, make object detection more accurate. Our framework has the unique ability to: i) estimate camera poses only from object detections, ii) enhance camera pose estimation, compared to feature-point-based SFM algorithms, iii) improve object detections given multiple uncalibrated images, compared to independently detecting objects in single images. Extensive quantitative results on three datasets – LiDAR cars, street-view pedestrians, and Kinect office desktop – verify our theoretical claims.

## 1 Introduction

Joint object recognition and 3D reconstruction of complex scenes from images is one of the critical capabilities of an intelligent visual system. Consider the photographs in Figure 1(a). These show the same environment observed from a handful of viewpoints. Even if this is the first time you (the observer) have seen this environment, it is not difficult to infer: i) the spatial structure of the scene and the way objects are organized in the physical space; ii) the semantic content of the scene and its individual components. State-of-the-art methods for object recognition [9,21,10,20] typically describe the scene with a list of class labels (e.g.



**Fig. 1.** Main objective of SSFM. (a) Input photos showing the same environment observed from a handful of viewpoints. (b) Traditional object recognition algorithms identify objects in 2D without reasoning about the 3D geometry. (c) SFM returns 3D scene reconstruction (3D point clouds) with no semantic information attached to it. (d) SSFM aims to jointly recognize objects and reconstruct the underlying 3D geometry of the scene (cameras, points and objects).

a chair, a desk, etc...) along with their 2D location and scale, but are unable to account for the 3D spatial structure of the scene and object configurations (Figure 1(b)). On the other hand, reconstruction methods (e.g. those based on SFM) [26,8,31,24,32] produce metric recovery of object and scene 3D structure (3D point clouds) but are mostly unable to infer the semantic content of its components (Figure 1(c)).

In this chapter we seek to fill this representation gap and propose a new framework for jointly recognizing objects as well as discovering their spatial organization in 3D (Figure 1(d)). The key concept we explore in this work is that measurements across viewpoints must be semantically and geometrically consistent. By measurements, we refer to the set of objects that can be detected in the image (e.g. a chair or monitor in Figure 1), their  $x,y$  location in the image, their scale (approximated by a bounding box) and their pose. Given a set of measurements from one view point, we expect to see a set of corresponding measurements (up to occlusions) from different view points which must be consistent with the fact that the view point has changed. For instance, the chair in Figure 1(a) appears in two views and its location, scale and pose variation across the two views must be consistent with the view point transformation. In this work we exploit this property and introduce a novel joint probability model where object detection and 3D structure estimation are solved in a coherent fashion.

Our proposed method has the merit of enhancing both 3D reconstruction and visual recognition capabilities in two ways: i) *Enhancing 3D reconstruction*: Our framework can help overcome a crucial limitation of scene/object modeling methods. State-of-the-art SFM techniques mostly fail when dealing with challenging camera configurations (e.g. when the views are too few and the view baseline is too large). This failure occurs as it is very hard to establish correct feature correspondences for widely separated views. For instance, the 3D reconstruction in Figure 1(c) was obtained using a state-of-the-art SFM algorithm [13] using 43 densely-sampled pictures of an office. The same algorithm would not work if we just used the two images in Figure 1(a) for the reasons mentioned above. By reasoning at the semantic level, and by establishing object correspondences

across views, our framework creates the conditions for overcoming this limitation. We show that our framework has the ability to estimate camera poses from object detections only. Moreover, our framework can still exploit traditional SFM constraints based on feature correspondences to make the 3D reconstruction process robust. We show that our method can significantly outperform across-view feature matching SFM algorithms such as [31,23] (Table 1). ii) *Enhancing visual recognition*: Traditional recognition methods are typically prone to produce false alarms when appearance cues are not discriminative enough and no contextual information about the scene is available. For instance, the cabinet in Figure 1(a) can be easily confused with a monitor as they both share similar appearance characteristics. By reasoning at the geometrical level, our framework is able to identify those hypotheses that are not consistent with the underlying geometry and reduce their confidence score accordingly. Our model leads to promising experimental results showing improvements in object detection rates compared with the state-of-the-art methods such as [9] (Figure 7 and Table 2). Also, we show that we can automatically establish object correspondence across views.

## 2 Related Works

Recently, a number of approaches have explored the idea of combining semantic cues with geometrical constraints for scene understanding. Notable examples are [14,30,22,33,17]. These focus on single images and, unlike our work, they do not attempt to enforce consistency across views. Moreover, they make restrictive assumptions on the camera and scene configuration. Other methods have been proposed to recognize objects with multi-view geometry [19,16], but they assume that the underlying scene geometry is available. A large number of works have proposed solutions for interpreting complex scenes from 3D data [11,18,28,27] or a combination of 3D data and imagery [3]. However, in most of these methods 3D information is either provided by external devices (e.g. 3D scanning systems such as LiDAR) or using traditional SFM techniques. In either case, unlike our framework, the recognition and reconstruction steps are separated and independent. [5] attempts joint estimation using a “cognitive loop” but requires a dedicated stereo-camera architecture and makes assumptions about camera motion. Having our preliminary result published as [1], we are the first to make these two steps coherent within a setting that requires only images with uncalibrated cameras (up to internal parameters) and arbitrary scene-camera configurations.

## 3 The Semantic Structure from Motion Model

Conventional rigid *structure from motion* (SFM) addresses the problem of recovering camera parameters  $\mathbf{C}$  and the 3D locations of scene points  $\mathbf{Q}$ , given observed 2D image feature points. In this chapter, we propose a new formulation where, in addition to the geometrical constraints provided by SFM, we

take advantage of both the semantic and geometrical properties associated with objects in the scene in order to recover  $\mathbf{C}$  and  $\mathbf{Q}$  as well as the 3D locations, poses, and category memberships of objects  $\mathbf{O}$  in the scene. We call this *semantic structure from motion* (SSFM). The key intuition is that, in addition to image features, the measurements of objects across views provides additional geometrical constraints that relate camera and scene parameters. We formulate SSFM as a maximum likelihood estimation (MLE) problem whose goal is to find the best configuration of cameras, 3D points and 3D objects that are compatible with the measurements provided by a set of images.

### 3.1 Problem Formulation

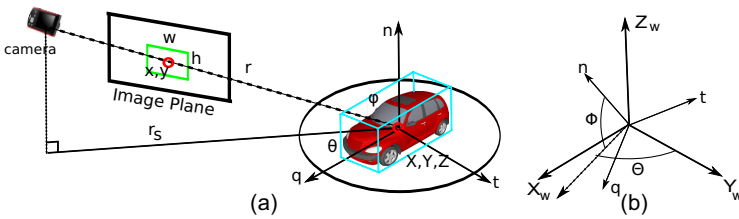
In this section we define the SSFM problem and formulate it as an MLE problem. We first define the main variables involved in SSFM, and then discuss the MLE formulation.

**Cameras.** Let  $\mathbf{C}$  denote the camera parameters.  $\mathbf{C} = \{C^k\} = \{K^k, R^k, T^k\}$  where  $K$  is the camera matrix capturing the internal parameters,  $R$  rotation matrix, and  $T$  translation vector with respect to a common world reference system.  $K$  is assumed to be known, whereas  $\{R, T\}$  are *unknown*. Throughout this chapter, the camera is indexed by  $k$  as a superscript.

**3D Points  $\mathbf{Q}$  and Measurements  $\mathbf{q}, \mathbf{u}$ .** Let  $\mathbf{Q} = \{Q_s\}$  denote a set of 3D points  $Q_s$ . Each 3D point  $Q_s$  is specified by  $(X_s, Y_s, Z_s)$  describing the 3D point location in the world reference system.  $\mathbf{Q}$  is an *unknown* in our problem. Denote by  $\mathbf{q} = \{q_i^k\}$  the set of point *measurements* (image features) for all the cameras. Namely,  $q_i^k$  is the  $i^{\text{th}}$  point measurement in image (camera)  $k$ . A point measurement is described by the measurement vector  $q^k = \{x, y, a\}_i^k$ , where  $x, y$  describe the point image location, and  $a$  is a local descriptor that captures the local neighborhood appearance of the point in image  $k$ . These measurements may be obtained using feature detectors and descriptors such as [23,35]. Since each image measurement  $\{q_i^k\}$  is assumed to correspond to a certain physical 3D point  $Q_s$ , we model such correspondence by introducing an indicator variable  $u_i^k$ , where  $u_i^k = s$  if  $q_i^k$  corresponds to  $Q_s$ . A similar notation was also introduced in [7]. A set of indicator variables  $\mathbf{u} = \{u_i^k\}$  allows us to establish feature correspondences across views and to relate feature matches with 3D point candidates (Section 3.3). Unlike [7], we assume the feature correspondences can be measured by feature matching algorithms such as [23]. Throughout this chapter,  $Q$  and  $q$  are indexed by  $s$  and  $i$  respectively and they appear as subscripts.

**3D Objects  $\mathbf{O}$  and Measurements  $\mathbf{o}$ .** Let  $\mathbf{O} = \{O_t\}$  denote a set of 3D objects  $O_t$ . As Figure 2 illustrates, the  $t^{\text{th}}$  3D objects  $O_t$  is specified by a 3D location  $(X_t, Y_t, Z_t)$ , a pose  $(\Theta_t, \Phi_t)$ , and a category label  $c_t$  (e.g. *car*, *person*, etc...). Thus, a 3D object is parametrized by  $O_t = (X, Y, Z, \Theta, \Phi, c)_t$ . The set  $\mathbf{O}$  is an *unknown* in our problem. Denote by  $\mathbf{o} = \{o_j^k\}$  the set of object *measurements* for all the cameras. Thus,  $o_j^k$  is the  $j^{\text{th}}$  measurement of an object in image (camera)  $k$ . An object measurement is described by the following measurement

vector  $o_j^k = \{x, y, w, h, \theta, \phi, c\}_j^k$  (Figure 2). As discussed in Section 3.2, these measurements may be obtained using any state-of-the-art object detector that can return the probability that certain location  $x, y$  in an image is occupied by an object with category  $c$ , scale  $h, w$ , and pose  $\theta, \phi$  (e.g. [29]). Similar to the 3D point case, if an object measurement  $o_j^k$  in image  $k$  is assumed to correspond to some physical 3D object  $O_t$ , such correspondence may be modeled by introducing an indicator variable  $v_j^k$ , where  $v_j^k = t$  if  $o_j^k$  corresponds to 3D object  $O_t$ . For the object case, the correspondences are automatically obtained by projecting 3D object into the images (Section 3.2). Thus, from this point on, we assume 2D object observations are given by  $\mathbf{o}$ . We denote 3D object and 2D object using the subscript index  $t$  and  $j$  respectively.



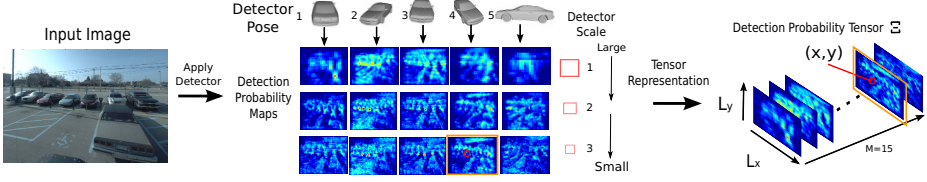
**Fig. 2.** 3D object’s location and pose parametrization. (a) Assume an object is enclosed by the tightest bounding cube. The object 3D location  $X, Y, Z$  is the centroid of the bounding cube (red circle). The object’s pose is defined by the bounding cube’s three perpendicular surface’s norms that are  $n, q, t$  and parametrized by the angles  $\Theta, \Phi$  in a given world reference system (b).  $r$  is the ray connecting  $O$  and the camera center. Let zenith angle  $\phi$  be the angle between  $r$  and  $n$ , and azimuth angle  $\theta$  be the angle between  $q$  and  $r_s$ , where  $r_s$  is the projection of  $r$  onto the plane perpendicular to  $n$ . Notice that we assume there is no in-plane rotation of the camera. We parametrize an object measurement in the image by the location  $x, y$  of tightest bounding box enclosing the object, the width  $w$  and height  $h$  of the bounding box (object 2D scale), the object pose  $\theta, \phi$ , and class  $c$ .

**MLE Formulation.** Our goal is to estimate a configuration of  $\mathbf{Q}, \mathbf{O}$  and  $\mathbf{C}$  that is consistent with the feature point measurements  $\mathbf{q}, \mathbf{u}$  and the object measurements  $\mathbf{o}$ . We formulate this estimation as the one of finding  $\mathbf{Q}, \mathbf{O}, \mathbf{C}$  such that the joint likelihood is maximized:

$$\begin{aligned} \{\mathbf{Q}, \mathbf{O}, \mathbf{C}\} &= \arg \max_{\mathbf{Q}, \mathbf{O}, \mathbf{C}} \Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{O}, \mathbf{C}) \\ &= \arg \max_{\mathbf{Q}, \mathbf{O}, \mathbf{C}} \Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) \Pr(\mathbf{o} | \mathbf{O}, \mathbf{C}) \end{aligned} \tag{1}$$

where the last expression is obtained by assuming that, given  $\mathbf{C}, \mathbf{Q}$  and  $\mathbf{O}$ , the measurements associated with 3D objects and 3D points are conditionally

<sup>1</sup> State of the art object detectors such as [9] can be modified so as to enable pose classification, as discussed in Section 5.1.



**Fig. 3.** Multi-pose and multi-scale object detection illustration. The “probability maps” are obtained by applying car detector with different scales and poses on the left image. The color from red to deep blue indicates the detector response from high to low. We used LSVM [9] (Section 5.1) to obtain these probability maps. In this example,  $\Xi$  has dimensions  $L_x \times L_y \times 15$ . If the scale=3 (small), pose=4, and category=car,  $\Pi$  will return the index  $\pi = 14$  (the red circle). Thus,  $\Xi(x, y, 14)$  will return the confidence of detecting a car at small scale and pose=4 at location  $x, y$  in the image (the orange rectangle).

independent. In the next two sections we show how to estimate the two likelihood terms  $\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$  (Equation 4 or 5) and  $\Pr(\mathbf{o} | \mathbf{O}, \mathbf{C})$  (Equation 3).

### 3.2 Object Likelihood $\Pr(\mathbf{o} | \mathbf{O}, \mathbf{C})$

$\Pr(\mathbf{o} | \mathbf{O}, \mathbf{C})$  measures the likelihood of object measurements  $\mathbf{o}$  given the camera and object configurations  $\mathbf{O}, \mathbf{C}$ . This term can be estimated by computing the *agreement* between predicted measurements and actual measurements. Predicted measurements are obtained by introducing a mapping  $\omega_t^k = \omega^k(O_t) = \omega^k((X, Y, Z, \Theta, \Phi, c)_t)$  that relates the parameters describing the 3D object  $O_t$  to the image of camera  $C^k$ . Thus,  $\omega_t^k$  is a parameter vector that contains the predicted location, pose, scale and category of  $O_t$  in  $C^k$ . Next, we present expressions for predicting the measurements and relating them to actual measurements and for obtaining an estimate of the likelihood term.

**Computing Predicted Measurements.** The transformation  $\omega_t^k = \omega^k(O_t)$  can be computed once cameras  $\mathbf{C}$  are known. Specifically, let us denote by  $X_t^k, Y_t^k, Z_t^k$  the 3D location of  $O_t$  in the reference system of  $C^k$  and by  $\Theta_t^k, \Phi_t^k$  its 3D pose (these can be obtained from  $X_t, Y_t, Z_t, \Theta_t, \Phi_t$  in the world reference system by means of a (known) rigid transformation). Predicted location  $(x_t^k, y_t^k)$  and pose  $(\phi_t^k, \theta_t^k)$  of  $O_t$  in camera  $C^k$  can be computed by using the camera projection matrix [15] as  $[x_t^k, y_t^k, 1]' = K^k [X_t^k, Y_t^k, Z_t^k]' / Z_t^k$  and  $[\phi_t^k, \theta_t^k] = [\Phi_t^k, \Theta_t^k]$ . Predicting 2D object scales in the image requires a more complex geometrical derivation that goes beyond the scope of this chapter. We introduce an approximated simplified mapping defined as follows:

$$\begin{cases} w_t^k = f_k \cdot W(\Theta_t^k, \Phi_t^k, c_t) / Z_t^k \\ h_t^k = f_k \cdot H(\Theta_t^k, \Phi_t^k, c_t) / Z_t^k \end{cases} \quad (2)$$

where  $w_t^k, h_t^k$  denote the predicted object 2D scale (similar to Figure 2),  $f_k$  is the focal length of the  $k^{\text{th}}$  camera.  $W(\Theta_t^k, \Phi_t^k, c_t)$  and  $H(\Theta_t^k, \Phi_t^k, c_t)$  are learned

(scalar) mapping that describe the typical relationship between physical object bounding cube and object image bounding box. The equations above allow us to fully estimate the object prediction vector  $\omega_t^k = \{x, y, w, h, \phi, \theta, c\}_t^k$  for object  $O_t$  in camera  $C^k$ .

**Learning Object Size Mapping.**  $W(\Theta_t^k, \Phi_t^k, c_t)$  and  $H(\Theta_t^k, \Phi_t^k, c_t)$  are (scalar) mapping functions of the object pose  $\Theta_t^k, \Phi_t^k$  and category  $c_t$ . They can be learned by using ground truth 3D object bounding cubes and corresponding observations using ML regressor. The mappings  $W$  and  $H$  relate the physical object bounding cube with the  $t^{th}$  object bounding box size (parametrized by  $w_t$  and  $h_t$ ) in the image. In the validation set, we have 3D objects  $\{o_t\} = \{\tilde{w}_t, \tilde{h}_t, \tilde{Z}_t, \tilde{\Theta}_t, \tilde{\Phi}_t, c_t\}$  with ground truth scale  $\tilde{w}_t, \tilde{h}_t$ , depth  $\tilde{Z}_t$ , pose  $\tilde{\Theta}_t, \tilde{\Phi}_t$ , and category  $c_t$ . We formulate the scale likelihood as  $\Pr(W(\Theta_t, \Phi_t, c_t) | \tilde{w}_t) \propto \exp(-(f \cdot W(\tilde{\Theta}_t, \tilde{\Phi}_t, c_t) / \tilde{Z}_t - \tilde{w}_t)^2 / \sigma_w)$  and  $\Pr(H(\Theta_t, \Phi_t, c_t) | \tilde{h}_t) \propto \exp(-(f \cdot H(\tilde{\Theta}_t, \tilde{\Phi}_t, c_t) / \tilde{Z}_t - \tilde{h}_t)^2 / \sigma_h)$ . Therefore, with the validation set,  $W$  and  $H$  can be learned as the mean value:

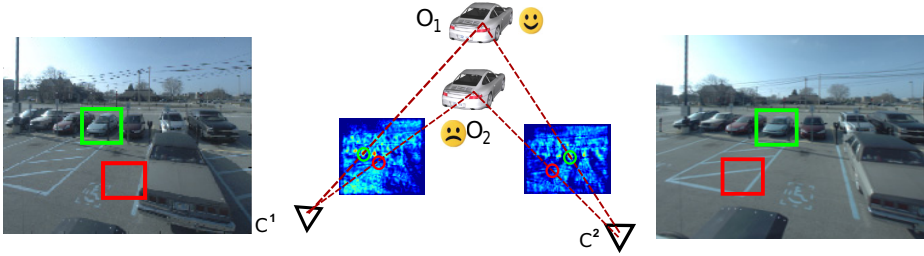
$$\begin{cases} W(\Theta, \Phi, c) = \frac{1}{N_t^*} \sum_{c_t=c, \tilde{\Theta}_t=\Theta, \tilde{\Phi}_t=\Phi} \tilde{w}_t \cdot \tilde{Z}_t / f \\ H(\Theta, \Phi, c) = \frac{1}{N_t^*} \sum_{c_t=c, \tilde{\Theta}_t=\Theta, \tilde{\Phi}_t=\Phi} \tilde{h}_t \cdot \tilde{Z}_t / f \end{cases}$$

where  $N_t^*$  is number of objects that have the pose as  $\Phi, \Theta$  and category  $c$ .

**Measurements as Probability Maps.**  $\Pr(o|O, C)$  can be now estimated by computing the *agreement* between predicted measurements and actual measurements. Such agreement is readily available using the set of probability values returned by object detectors such as [9] applied to images (Figure 3). The output of this detection process for the image of  $C^k$  is a tensor  $\Xi^k$  of  $M$  probability maps wherein each map captures the likelihood that an object of category  $c$  with scale  $w, h$  and pose  $\theta, \phi$  presents at location  $x, y$  in the image. Thus, we can interpret  $\Xi^k$  as one  $L_x \times L_y \times M$  tensor, where  $L_x$  and  $L_y$  are the image width and height and  $M$  adds up to the number of object categories, scales and poses. Let us denote by  $\Pi : \{w, h, \phi, \theta, c\} \rightarrow \pi \in 1 \dots M$  the *indexing function* that allows retrieval from  $\Xi^k$  the detection probability at any location  $x, y$  given a set of values for scale, pose and category. Figure 3 shows an example of a set of 15 probability maps for only one object category (i.e., the *car* category), three scales and five poses associated with a given image. Notice that since measurements can be extracted directly from  $\Xi^k$  once the mapping 3D-object-image  $\omega$  is computed (Figure 4), the 2D objects of the  $k^{th}$  image are automatically associated with the 3D objects. As a result, across-view one-to-one object correspondences are also established.

**Estimating the Likelihood Term.** The key idea is that the set  $\Xi^k$  of probability maps along with  $\pi$  can be used to estimate  $\Pr(o|O, C)$  given the predicted measurements. To illustrate this, let us start by considering an estimation of the likelihood term  $\Pr(o|O_t, C^k)$  for  $O_t$  observed from camera  $C^k$ . Using  $\omega_t^k$ , we can predict the object's scale  $\{w, h\}_t^k$ , pose  $\{\phi, \theta\}_t^k$  and category  $c_t^k$ . This allows us to retrieve from  $\Xi^k$  the probability of detecting an object at the predicted location  $\{x, y\}_t^k$  by using the indexing function  $\pi_t^k$ , and in turn estimate





**Fig. 4.** Mapping 3D objects to measurements. In this example, the measurements of  $O_1$  (green) correspond to high value location in the probability maps, while the 2D measurements of  $O_2$  (red) correspond to low value location in the probability maps. Therefore,  $\Pr(\mathbf{o}|O_1, \mathbf{C})$  is much higher than  $\Pr(\mathbf{o}|O_2, \mathbf{C})$ .

$\Pr(\mathbf{o}|O_t, C^k) = \Xi^k(x_t^k, y_t^k, \pi(w_t^k, h_t^k, \phi_t^k, \theta_t^k, c_t^k))$ . Assuming that objects are independent from each other and camera configurations are independent, the joint likelihood of objects and cameras can be approximated as:

$$\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C}) \propto \prod_t^{N_t} \Pr(\mathbf{o}|O_t, \mathbf{C}) \propto \prod_t^{N_t} (1 - \prod_k^{N_k} (1 - \Pr(\mathbf{o}|O_t, C^k))) \quad (3)$$

where  $N_t$  is the number of objects and  $N_k$  is the number of cameras.  $N_t$  is in general unknown, but it can be estimated using detection probability maps (Section 4.1). Notice that this term does not penalize objects that are observed only by a portion of images while they are truncated or occluded in other images.  $\Pr(\mathbf{o}|O_t, \mathbf{C})$  is only partially affected by an occluded or truncated object  $O_t$  in the  $k^{\text{th}}$  image even if the object leads to a low value for  $\Pr(\mathbf{o}|O_t, C^k)$ .

### 3.3 Points Likelihood $\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$

$\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$  measures the likelihood of the 3D points and cameras given the measurements of 3D points and their correspondences across views. This likelihood term can be estimated by computing the *agreement* between predicted measurements and actual measurements. Similar to the 3D object case, predicted measurements are obtained by introducing a mapping from 3D points to the images.

**Predicted Measurements.** Predicted measurements can be easily obtained once the cameras  $\mathbf{C}$  are known. We indicate by  $q_s^k$  the predicted measurement (a pixel location in the image) of the  $s^{\text{th}}$  point  $Q_s$  in camera  $C^k$ .  $q_s^k$  can be obtained by using the projection matrix of camera  $C^k$ . Since we know which point is being projected, we have a prediction for the indicator variable  $u$  as well.

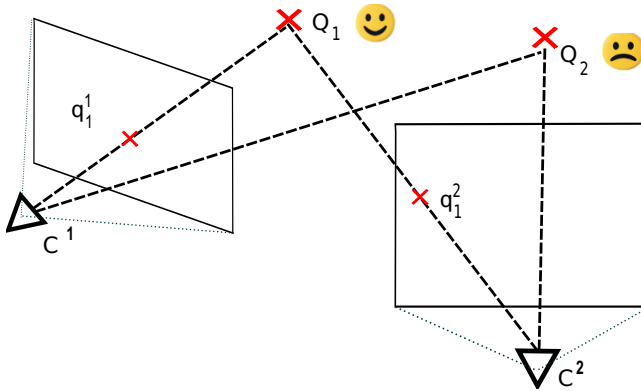
**Point Measurements.** Point measurements are denoted by  $q_i^k = \{x, y, a\}_i^k$ , where  $x, y$  describe the point location in image  $k$  of measurement  $i$ , and  $a$  is a local descriptor that captures the local appearance of the point in a neighborhood

of image  $k$ . We obtain location measurements  $\{x, y\}_i^k$  using a DOG detector equipped with a SIFT descriptor for estimating  $a_i^k$  [23]. Measurements for feature correspondences (matches) across images are obtained by matching the point features.

**Estimating the Likelihood Term.**  $\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$  can be estimated by computing the agreement between predicted measurements and the actual measurements (Figure 5). Let us start by considering the likelihood term  $\Pr(q | Q_s, C^k)$  for one point  $Q_s$  and for camera  $C^k$ . As introduced in [7], one possible strategy for computing such agreement assumes that the location of measurements and predictions are equal up to a noise  $n$  - that is,  $q_i^k = q_s^k + n$ , where  $s = u_i^k$ . If we assume zero mean Gaussian noise, we can estimate  $\Pr(q_i^k | Q_s, C^k) \propto \exp(-(q_i^k - q_{u_i^k}^k)^2 / \sigma_q)$ , leading to the following expression for the likelihood:

$$\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) = \prod_i^{N_Q} \prod_k^{N_k} \exp(-(q_i^k - q_{u_i^k}^k)^2 / \sigma_q) \tag{4}$$

where  $N_k$  is the number of cameras,  $N_Q$  is the number of points, and  $\sigma_q$  is the variance of 2D point projection measurement error. This is obtained by assuming independence among points and among cameras.



**Fig. 5.** Estimating the likelihood term for points.  $q_1^1$  and  $q_1^2$  are point measurements.  $Q_1$  and  $Q_2$  are candidate 3D points corresponding to  $q_1^1$  and  $q_1^2$ . In this case, the likelihood of  $Q_1$  is higher than  $Q_2$ , because the projections of  $Q_1$  are closer to the measurements.

We also propose an alternative estimator for  $\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$ . While this estimator leads to a coarser approximation for the likelihood, it makes the inference process more efficient and produces more stable results. This estimator exploits the epipolar constraints relating camera pairs. Given a pair of cameras  $C^l$  and  $C^k$ , we can estimate the fundamental matrix  $F_{l,k}$ . Suppose  $q_i^k, q_j^l$  are from  $C^k$  and  $C^l$  respectively, and the matching algorithm predicts that  $q_i^k$  and  $q_j^l$  are in correspondence.  $F_{l,k}$  can predict the epipolar line  $\xi_i^{l,k}$  (or  $\xi_j^{k,l}$ ) of  $q_i^k$  (or  $q_j^l$ ) in

image  $C^l$  (or  $C^k$ ). If we model the distance  $d_{j,i}^{l,k}$  between  $\xi_i^{l,k}$  and  $q_j^l$  as zero-mean Gaussian with variance  $\sigma_u$ ,  $\Pr(q_i^k, q_j^l | Q_s, C_l, C_k) \propto \exp(-d_{j,i}^{l,k}/\sigma_u)$ . Notice that this expression does not account for appearance similarity between matched features – that is the similarity between the descriptors  $a_i^k$  and  $a_j^l$ . We model appearance similarity as  $\exp(-\frac{\alpha(a_i^k, a_j^l)}{\sigma_\alpha})$  where  $\alpha(\cdot, \cdot)$  captures the distance between two feature vectors and  $\sigma_\alpha$  the variance of the appearance similarity. Overall, we obtain the following expression for the likelihood term:

$$\begin{aligned} \Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) &\propto \prod_{k \neq l}^{N_k} \prod_{i \neq j}^{N_s} \Pr(q_i^k, q_j^l | Q_s, C_l, C_k) \\ &\propto \prod_{k \neq l}^{N_k} \prod_{i \neq j}^{N_s} \exp\left(-\frac{d_{j,i}^{l,k}}{\sigma_u}\right) \exp\left(-\frac{\alpha(a_i^k, a_j^l)}{\sigma_\alpha}\right) \end{aligned} \quad (5)$$

Equation 5 is obtained by assuming that feature locations and appearance are independent. During the learning stage, we learn the variance  $\sigma_u$  and  $\sigma_\alpha$  using an ML estimator on a validation set. Notice that  $\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$  is no longer a function of  $Q_s$ . Hence, during every iterations in Algorithm. 1, we can avoid estimating 3D points, which is usually an expensive process (e.g. see the bundle adjustment algorithm [34]). This significantly reduces the complexity for solving the MLE problem.

## 4 Max-Likelihood Estimation with Sampling

Our goal is to estimate camera parameters, points, and objects so as to maximize Equation 1. Due to the high dimensionality of the parameter space, we propose to sample  $\mathbf{C}, \mathbf{Q}, \mathbf{O}$  from  $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$  similar to [7]. This allows us to approximate the distribution of  $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$  and find the  $\mathbf{C}, \mathbf{Q}, \mathbf{O}$  that maximize the likelihood. In Section 4.1 we discuss the initialization of the sampling process, and in Section 4.2 we describe a modified formulation of the Markov Chain Monte Carlo (MCMC) sampling algorithm for solving the MLE problem.

### 4.1 Parameter Initialization

Appropriate initialization of cameras, objects, and points is a critical step in the sampling method. We initialize camera configurations (i.e. estimate camera configurations that are geometrically compatible with the observations) using feature point matches and object detections.

**Camera Initialization by Feature Points.** We follow [24] to initialize (estimate)  $\mathbf{C}$  from image measurements  $\mathbf{q}$ . Due to the metric reconstruction ambiguity, we scale the estimated camera translation with several random values to obtain several camera pose initializations.

<sup>2</sup> To account for outliers, we set a threshold on  $d_{j,i}^{l,k}$ . Namely, if  $\bar{d}_{j,i}^{l,k}$  is the measurement, we set  $d_{j,i}^{l,k} = \min(\bar{d}_{j,i}^{l,k}, \Gamma)$ . We learn the outlier threshold  $\Gamma$  using a validation set.

**Camera Initialization by Objects.** We use a standard object detector [9] to detect 2D objects and estimate object pose and scale (Section 5.1). Next, we use these object detections to form possible object correspondences and use these to estimate several possible initial camera configurations. Assume the  $k^{\text{th}}$  camera has a set of object detections  $\mathbf{o}^k = \{o_t^k\}$ , where  $o_t^k$  is the  $t^{\text{th}}$  detected 2D object in the  $k^{\text{th}}$  camera.  $o_t^k$  captures the 2D object location  $x_t^k, y_t^k$  and bounding box scale  $w_t^k, h_t^k$  (i.e.  $o_t^k = \{x_t^k, y_t^k, w_t^k, h_t^k\}$ ). If the object detector has the ability to classify object pose,  $o_t^k$  also captures the pose  $\phi_t^k, \theta_t^k$  (i.e.  $o_t^k = \{x_t^k, y_t^k, w_t^k, h_t^k, \phi_t^k, \theta_t^k\}$ ). Depending on whether the pose  $\phi_t^k, \theta_t^k$  and the pre-learned object scale  $W, H$  (so as to allow us to use Equation 2 to compute the object depth) are used or not, there are three ways to initialize the camera extrinsic parameters  $R^k, T^k$  (the intrinsic parameter  $K^k$  is known): 1) initialize cameras by only using object scale  $W, H$ ; 2) initialize cameras by only using object pose  $\phi, \theta$ ; 3) initialize cameras by using scale  $W, H$  and pose  $\phi, \theta$ . In our experiments, case 1 applies on the pedestrian dataset, as the pose cannot be robustly estimated for pedestrians; case 2 does not apply on any of our experiments; case 3 applies on the Ford car dataset and the office dataset. The propositions in Section 7 give necessary conditions for estimating the camera parameters. These propositions establish the least number of objects that are necessary to be observed for each of the initialization cases above. These propositions also give conditions for estimating camera parameters given a number of object detections. Based on a list of possible object correspondences across images, these propositions can be used for generating hypotheses for camera and object configurations for initializing the sampling algorithm.

**Points and Objects Initialization.** Camera configurations obtained by using points and objects form the initialization set. For each of these initial configurations, object detections are used to initialize objects in 3D using the mapping in Equation 2. If certain initialized 3D objects are too near to others (location and pose-wise), they are merged to a single one. We use the distance between different initializations to remove overlapping 3D initializations. Suppose that, after the initializations, the objects are  $\{O_t\} = \{X_t, Y_t, Z_t, \Phi_t, \Theta_t, c_t, \rho_t\}$  where  $X_t, Y_t, Z_t$  is the object coordinates in the world coordinate system,  $\Phi_t, \Theta_t$  is the object pose in the world coordinate system,  $c_t$  is the object category, and  $\rho_t$  is the 2D detection probability of the 2D object that initializes  $O_t$ . We perform a greedy search to remove the overlapping object:  $O_t$  will be removed from the 3D object set if there is another object  $O_s$  with  $c_s = c_t$  and  $\rho_s > \rho_t$  so that  $||[X_t, Y_t, Z_t] - [X_s, Y_s, Z_s]|| < t_{XYZ}$  and  $||[\Phi_t, \Theta_t] - [\Phi_s, \Theta_s]|| < t_{\Phi\Theta}$ . The threshold  $t_{XYZ}$  and  $t_{\Phi\Theta}$  are learned from a validation set where the ground truth object 3D location and pose are available. Similar to objects, for each camera configuration, feature points  $\mathbf{q}$  are used to initialize 3D points  $\mathbf{Q}$  by triangulation [15]. Correspondences between  $\mathbf{q}$  and  $\mathbf{Q}$  are established after the initialization. We use index  $r$  to indicate one out of  $R$  possible initializations for objects, cameras and points  $(\mathbf{C}_r, \mathbf{O}_r, \mathbf{Q}_r)$ .

## 4.2 Sample and Maximize the Likelihood

We sample  $\mathbf{C}, \mathbf{O}, \mathbf{Q}$  from the underlying  $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$  using a modified Metropolis algorithm [12] (Algorithm 1). Since the goal of the sampling is to identify a maximum, the samples should occur as near to  $\max \Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$  as possible, so as to increase the efficiency of the sampling algorithm. Thus, we only randomly sample  $\mathbf{C}$ , while the best configuration of  $\mathbf{O}$  and  $\mathbf{Q}$  given the proposed  $\mathbf{C}$  are estimated during each sampling step. In step 3, the estimation of  $\mathbf{O}'$  is obtained by greedy search within a neighborhood of the objects proposed during the previous sampling step (Section 4.3). Since the object detection scale and pose are highly quantized, the greedy search yields efficient and robust results in practice. In step 4, the estimation of  $\mathbf{Q}$  is based on the minimization of the projection error (Section 4.4).

By Algorithm 1, we can generate the sample  $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$  from the  $r^{\text{th}}$  initialization. From all of the samples, we estimate the maximum of  $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$  as follows. We concatenate  $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$  from different initializations into one sample point set  $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}$ . Next, the frequency of the samples will provide an approximation of the distribution of  $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$ . To identify the maximum, the MeanShift algorithm [4] is employed to cluster the samples. The center of the cluster with the highest sample number is regarded as the approximation of the maximum of  $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$  and thus taken as the solution of the final estimation of  $\mathbf{C}, \mathbf{O}, \mathbf{Q}$ .

Algorithm 1 can be applied using either Equation 4 or 5. If Equation 5 is used, the estimation is greatly simplified as  $\mathbf{Q}$  no longer appears in the optimization process. Hence step 4 is no longer required. Our experiments use the latter implementation.

---

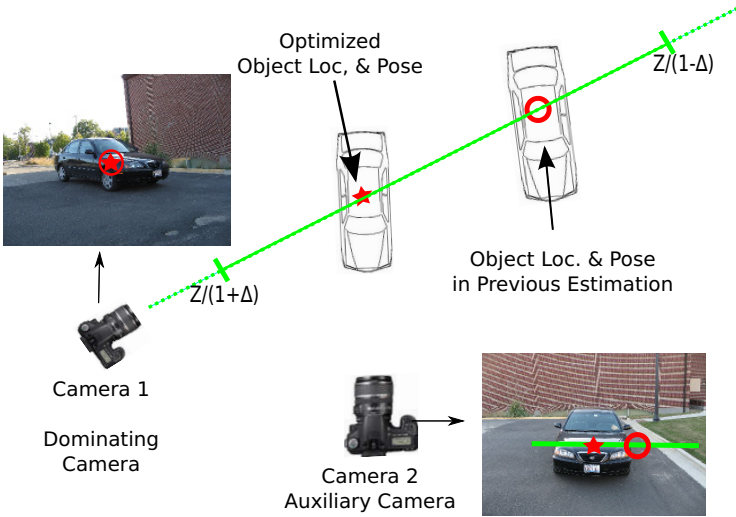
**Algorithm 1.** MCMC sampling from  $r^{\text{th}}$  initialization.

---

- 1: Start with  $r^{\text{th}}$  proposed initialization  $\mathbf{C}_r, \mathbf{O}_r, \mathbf{Q}_r$ . Set counter  $v = 0$ .
  - 2: Propose new camera parameter  $\mathbf{C}'$  with Gaussian probability whose mean is the previous sample and the co-variance matrix is uncorrelated.
  - 3: Propose new  $\mathbf{O}'$  within the neighborhood of previous object's estimation to maximize  $\Pr(\mathbf{o} | \mathbf{O}', \mathbf{C}')$ .
  - 4: Propose new  $\mathbf{Q}'$  with  $\mathbf{C}'$  to minimize the point projection error.
  - 5: Compute the acceptance ratio  $\alpha = \frac{\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{C}', \mathbf{O}', \mathbf{Q}')}{\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{C}, \mathbf{O}, \mathbf{Q})}$
  - 6: If  $\alpha \geq \rho$  where  $\rho$  is a uniform random variable  $\rho \sim U(0, 1)$ , then accept  $(\mathbf{C}, \mathbf{O}, \mathbf{Q}) = (\mathbf{C}', \mathbf{O}', \mathbf{Q}')$ . Record  $(\mathbf{C}, \mathbf{O}, \mathbf{Q})$  as a sample in  $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$ .
  - 7:  $v = v + 1$ . Goto 2 if  $v$  is smaller than the predefined max sample number; otherwise return  $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$  and end.
- 

## 4.3 Proposing New Objects

The goal of step 3 of Algorithm 1 is to propose and select the best  $\mathbf{O}'$  given newly proposed cameras  $\mathbf{C}'$ . Let us denote by  $\mathbf{O}(\mathbf{C})$  the estimation of the



**Fig. 6.** Proposing object candidates given newly proposed cameras. The red circle is the result of the estimation from step  $i$ . The green line collects the proposed 3D locations of the object centroid (i.e. a proposed line of sight). The estimation of the object in step  $i + 1$  is obtained as function of the image measurements, and shown as red star.

configuration of the objects (cameras) at MCMC sampling iteration  $i$ , and by  $\mathbf{O}'$  ( $\mathbf{C}'$ ) the configuration of objects (cameras) at the next sampling iteration.

**Proposing  $\mathbf{O}'$ :** Since the objects are assumed to be independent to each other, we focus on the single object  $O'_t$ . We propose a set of object candidates (locations, poses, scales) for  $O'_t$ , and we denote such set of candidates by  $\Psi_{\mathbf{C}'}(O_t)$ .  $\Psi_{\mathbf{C}'}(O_t)$  is obtained by sampling in the neighborhood in the parameter space of  $O_t$ . Without loss of generality, assume that the  $k^{th}$  image has the largest single-image detection likelihood for  $O_t$  given  $\mathbf{C}$ , i.e.  $\Pr(\mathbf{o}|O_t, C^k) = \max_{h=1 \dots N_k} \Pr(\mathbf{o}|O_t, C^h)$ . We define  $C^k$  as the “dominating camera” of  $O_t$  (Figure 6).  $o_t^k$  is the projection of  $O_t$  onto the  $k^{th}$  image. As a result of previous optimization,  $o_t^k$  is corresponding to the local maximum of 2D object detection probability. To increase the computing efficiency, we enforce that the proposed candidate of  $O'_t$  will generate a projection  $o_t'^k$  in image  $k$  that belongs to a neighborhood of  $o_t^k$ . More specifically, we enforce  $|o_t'^k - o_t^k| < \Delta o$  where  $\Delta o = \{\Delta x, \Delta y, \Delta h, \Delta w, \Delta \theta, \Delta \phi\}$ . We also enforce that the proposed object depth to be within a finite range  $Z_t^k/(1 + \Delta) < Z_t'^k < Z_t^k/(1 - \Delta)$ . Such proposals for  $O'_t$  form the set  $\Psi_{\mathbf{C}'}(O_t)$ . In Figure 6, the green line corresponds to the “location” component of  $\Psi_{\mathbf{C}'}(O_t)$ .

**Selecting  $\mathbf{O}'$ :** Again, let us focus on the single object  $O'_t$ . The new  $O'_t$  is selected as the element in  $\Psi_{\mathbf{C}'}(O_t)$  that maximizes the object measurement likelihood:

$$O'_t = \arg \max_{O'_t \in \Psi_{\mathbf{C}'}(O_t)} \Pr(\mathbf{o}|O'_t, \mathbf{C}') \tag{6}$$

As a reminder, the computation of  $\Pr(o|O'_t, \mathbf{C}')$  is explained in Section 3.2. Given the limited number of proposals within  $\Psi_{\mathbf{C}'}(O'_t)$ , an exhaustive search is feasible and it is computationally cheap to select  $O'_t$  using Equation 6. Finally, by selecting new estimations for every objects, the new estimation for objects is obtained as  $\mathbf{O}' = \{O'_t\}$ .

#### 4.4 Proposing 3D Points

The goal of step 4 of Algorithm 1 is to propose and select the best  $\mathbf{Q}'$  given newly proposed cameras  $\mathbf{C}'$ . If Equation 4 is used, the goal of proposing the new  $\mathbf{Q}'$  is to maximize the points likelihood:

$$\begin{aligned} \mathbf{Q}' &= \arg \max_{\mathbf{Q}'} \prod_i^{N_Q} \prod_k^{N_k} \exp(-(q_i^k - q_{u_i^k}^k)^2 / \sigma_q) \\ &= \arg \min_{\mathbf{Q}'} \sum_i^{N_Q} \sum_k^{N_k} (q_i^k - q_{u_i^k}^k)^2 \end{aligned} \quad (7)$$

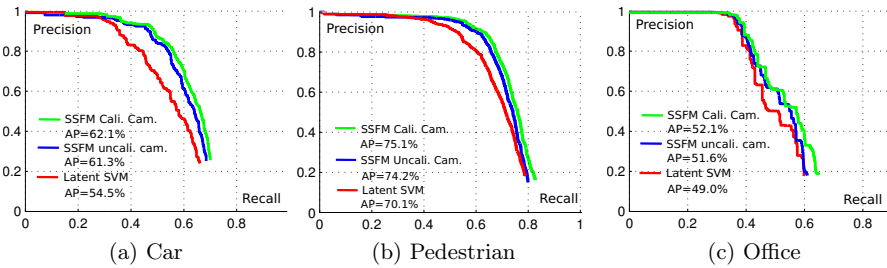
Notice that solving Equation 7 is equivalent to the objective function of bundle adjustment 34. Therefore, bundle adjustment can be applied given camera parameters to propose the new  $\mathbf{Q}'$  in Algorithm 1 step 4.

If Equation 5 is used, the 3D point likelihood  $\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$  is approximated using the epipolar geometry. Note  $\mathbf{Q}$  does not appear in Equation 5 and thus has no effect on the optimization process. The approximation gives the significant advantage of accelerating the sampling process Algorithm 1, since the optimization (bundle adjustment) of  $\mathbf{Q}$  is avoided. As a result,  $\mathbf{Q}$  is not estimated during the sampling process but is instead estimated by triangulation after the best camera configuration  $\mathbf{C}$  is found.

## 5 Evaluation

In this section we qualitatively demonstrate the ability of the SSFM model to jointly estimate the camera pose and improve the accuracy in detecting objects. We test SSFM on three datasets: the publicly available Ford Campus Vision and LiDAR Dataset 25, a novel Kinect office dataset 3, and a novel street-view pedestrian stereo-camera dataset. Anecdotal examples are shown in Figure 9. Although SSFM does not use any information from 3D points, the calibrated 3D points from LiDAR and Kinect allows us to easily obtain the ground truth information. The typical running time for one image pair with our Matlab single-thread implementation is  $\sim 20$  minutes. Benchmark comparisons with the state-of-the-art baseline detector *Latent SVM* 9 and point-based SFM approach *Bundler* 31 demonstrate that our method achieves significant improvement on object detection and camera pose estimation results.

<sup>3</sup> Available at <http://www.eecs.umich.edu/vision/projects/ssfm/index.html>



**Fig. 7.** Detection PR results by SSFM with calibrated cameras (green), SSFM with uncalibrated cameras (blue) and LSVM [9] (red). Figure 7c shows average results for mouse, keyboard and monitor categories. SSFM is applied on image pairs randomly selected from the testing set (unless otherwise stated). Calibration is obtained from ground truth.

To evaluate the object detection performance, we plot precision-recall (PR) curves and compare the average-precision (AP) value with baseline detector LSVM [9]. Object detection for SSFM is obtained by projecting the estimated 3D object bounding cube into each image. Given ground truth bounding boxes, we measure the object detection performance following the protocol of the PASCAL VOC Challenge[4]. LSVM baseline detector is applied to each image used by SSFM. Thus PR values are computed for each image for fair comparison.

To evaluate the camera pose estimate, we compare the camera pose estimation of SSFM with the state-of-the-art point-based structure-from-motion approach Bundler [31]. Bundler first employs the SIFT feature, five-points algorithm [24] and RANSAC to compute the fundamental matrix, and then applies Bundle Adjustment [34]. In certain configurations (e.g. wide baseline) RANSAC or Bundle Adjustment fail to return results. In such cases we take the camera pose estimation of five-points algorithm as the results for comparison. We follow the evaluation criteria in [24]. When comparing the camera pose estimation, we always assume the first camera to be at the canonical position. Denote  $R_{gt}$  and  $T_{gt}$  as the ground truth camera rotation and translation, and  $R_{est}$  and  $T_{est}$  the estimated camera rotation and translation. The error measurement of rotation  $e_R$  is the minimal rotating angle of  $R_{gt}R_{est}^{-1}$ . The error measurement of translation  $e_T$  is evaluated by the angle between the estimated baseline and the ground truth baseline, and  $e_T = \frac{T_{gt}^T R_{gt}^{-T} R_{est}^{-1} T_{est}}{|T_{gt}| \cdot |T_{est}|}$ . For a fair comparison, the error results are computed on the second camera.

We also analyze the performance of SSFM as a function of the number of cameras (views). A testing set is called  $N$ -view set if it contains  $M$  groups of  $N$  images. The testing sets with smaller number of views are first generated (i.e. 2-view set is the very first). If one  $N$ -view set is used, the  $N + 1$ -view testing set is generated by adding one additional random view to each of the  $M$  groups of  $N$  images.

<sup>4</sup> <http://pascallin.ecs.soton.ac.uk/challenges/VOC/>



## 5.1 Implementation Details

SSFm requires an object detector that is capable of determining the object pose. We use the state-of-the-art object detector [9] and treat object poses as extra-classes for each object category.

## 5.2 Ford Campus Vision Dataset [25]

The Ford Campus Vision dataset consists of images of cars aligned with 3D scans obtained using a LiDAR system. Ground truth camera parameters are also available. Our training / testing set contains 150 / 200 images of 4 / 5 different scenes. We randomly select 350 image pairs out of the testing images with the rule that every pair of images must capture the same scene. The training set for the car detector is the 3D object dataset [29]. This training set consists of 8 poses.

**Camera Pose Estimation:** SSFM obtains smaller translation estimation error than Bundler and comparable rotation estimation error (Table 1).

**Table 1.** Evaluation of camera pose estimation for two camera case.  $\bar{e}_T$  represents the mean of the camera translation estimation error, and  $\bar{e}_R$  the mean of the camera rotation estimation error.

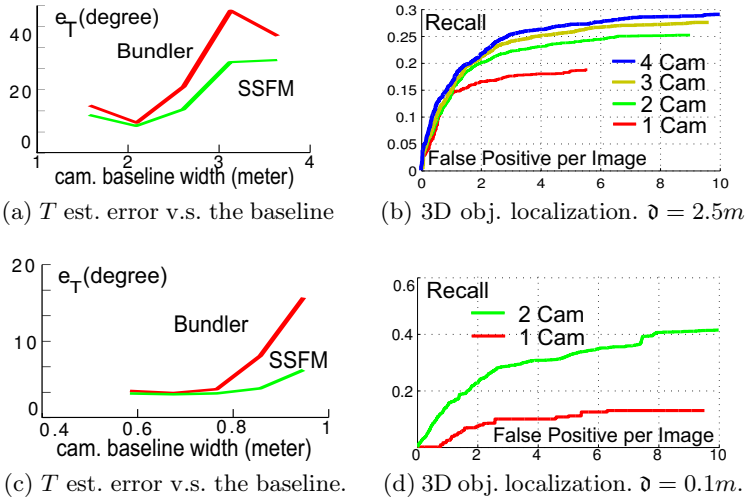
Dataset	$\bar{e}_T$ Bundler/SSFm	$\bar{e}_R$ Bundler/SSFm
Ford Campus Car	26.5/ <b>19.9</b> $^\circ$	$< 1^\circ / < 1^\circ$
Street Pedestrian	27.1/ <b>17.6</b> $^\circ$	21.1/ <b>3.1</b> $^\circ$
Office Desktop	8.5/ <b>4.7</b> $^\circ$	9.6/ <b>4.2</b> $^\circ$

**Table 2.** Camera pose estimation errors and object detection AP v.s. numbers of cameras on the Ford-car dataset. The baseline detector AP is 54.5%.

Camera #	2	3	4
Det. AP (Cali. Cam.)	62.1%	63.6%	64.2%
Det. AP (Uncali. Cam.)	61.3%	61.7%	62.6%
$\bar{e}_T$	19.9 $^\circ$	16.2 $^\circ$	13.9 $^\circ$

**Object Detection:** The PR by SSFM and the baseline detector are plotted in Figure 7a. Since ground truth annotation for small objects is difficult to obtain accurately, in this dataset we only test scales whose bounding box areas are larger than 0.6% of the image area. SSFM improves the detection precision and recall.

**Camera Baseline Width v.s. Pose Estimation:** We analyze the effect of baseline width on the camera pose estimation. Since the rotation estimations of



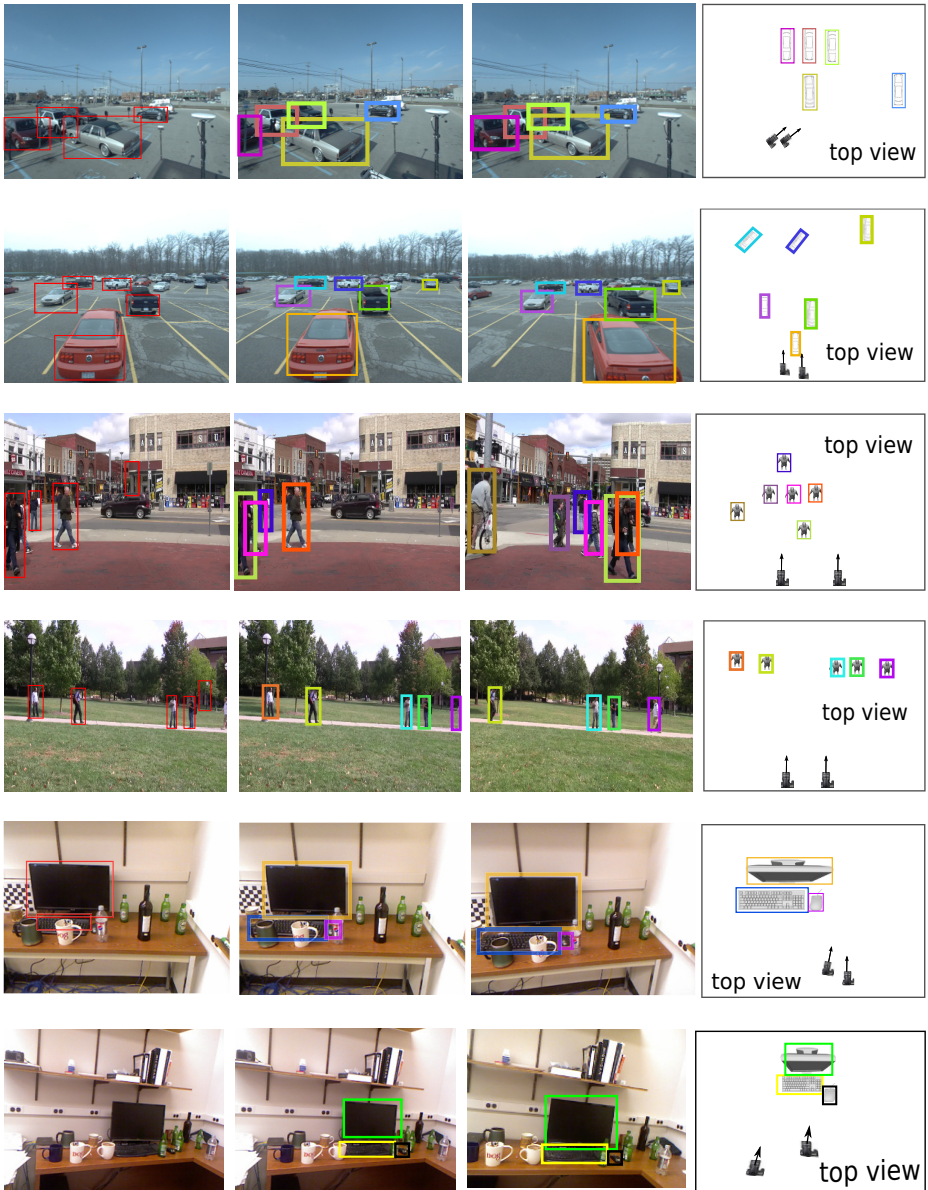
**Fig. 8.** System analysis of SSFM on Ford Car Dataset (a)(b) and Kinect Office Dataset (c)(d). For the car dataset, the typical object-to-camera distance is  $10 \sim 30$  meters. For the office dataset, the typical object-to-camera distance is  $1 \sim 2$  meters.

both Bundler and SSFM contain little error, we only show the translation estimation error v.s. camera baseline width (Figure 8a). This experiment confirms the intuition that a wider baseline impacts more dramatically the performance of methods based on low level feature matching than does on methods such as SSFM where higher level semantics are used.

**Comparison for Different Number of Cameras:** Table 2 shows the camera pose estimation error and the object detection AP as a function of the number of views (cameras) used to run SSFM. As more cameras are available, SSFM tends to achieve better object detection result and camera translation estimation.

**3D Object Localization Performance:** Due to the metric-reconstruction ambiguity, we use calibrated cameras in this experiment to enforce that the coordinates of 3D objects have a physical meaning. We manually label the 3D bounding boxes of cars on the LiDAR 3D point cloud to obtain the ground truth car 3D locations. We consider a 3D detection to be true positive if the distance between its centroid and ground truth 3D object centroid is smaller than a threshold  $\delta$  (see figure captions). The 3D object localization for one camera (single view) is obtained by using its 2D bounding box scale and location [2]. SSFM performance increases as the number of views grows (Figure 8b).

**Object-Based Structure from Motion:** We disable the feature point detection and matching, thus no 2D points are used (i.e. just maximize  $\Pr(\mathbf{o}|\mathbf{C}, \mathbf{O})$ ). For the two-view case, the detection AP increases from the baseline 54.5% to 55.2%, while the error of camera pose estimation is  $\bar{e}_T = 81.2^\circ$  and  $\bar{e}_R = 21.2^\circ$ . Notice that random estimation of the parameters would yield  $\bar{e}_T = 90^\circ$  and



**Fig. 9.** Anecdotal examples. Column 1: Baseline object detection in the 1<sup>st</sup> image; Column 2,3: the final joint object detections projected in the 1<sup>st</sup> and 2<sup>nd</sup> image; Column 4: the top view of the scene. Colors in the last three columns show the object correspondences established by SSFM.

$\bar{e}_R = 90^\circ$ . To the best of our knowledge, this is the first time SFM has been tested based only on high-level cues (objects) rather than low-level / middle-level cues (e.g. points, lines, or areas). Notice that the

### 5.3 Kinect Office Desktop Dataset

We use Microsoft’s Kinect to collect images and corresponding 3D range data of several static indoor office environments. The ground truth camera parameters are obtained by aligning range data across different views. We manually identify the locations of ground truth 3D object bounding cubes similarly to the way we process Ford dataset. The objects in this dataset are monitors, keyboards, and mice. The testing and training sets contain 5 different office desktop scenes respectively and each scenario has  $\sim 50$  images. From each scenario, we randomly select 100 image pairs for testing or training. SSFM performance is evaluated using the ground truth information and compared against baseline algorithms. We show these results as Figure 7c, Table II, Figure 8c, and Figure 8d. SSFM estimates camera poses more accurately than point-based SFM, and detects objects more accurately than single-image detection method.

### 5.4 Stereo Street-View Pedestrian Dataset

We collected this dataset by simultaneously capturing pairs of images of street-view pedestrians. The two cameras are pre-calibrated so that the ground-truth camera poses are measured and known. The object category in this dataset is pedestrian. The training set of object detector is INRIA pedestrian dataset [6] with no pose label. The two cameras are parallel and their relative distance is  $4m$ . The typical object-to-camera distance is  $5 \sim 10m$ . The training set contains 200 image pairs in 5 different scenes. The testing set contains 200 image pairs in 6 other scenes. SSFM attains smaller camera pose estimation error compared to Bundler (Table II) and better detection rates than LSVM (Figure 7b). Notice in this dataset the baseline width of the two cameras is fixed thus we cannot analyze the camera pose estimation error v.s. camera baseline width and cannot carry out experiments with multiple cameras.

## 6 Conclusion

This chapter presents a new paradigm called the semantic structure from motion for jointly estimating 3D objects, 3D points and camera poses from multiple images. We demonstrated that semantic structure from motion is capable of estimating camera poses more accurately than point-based structure-from-motion methods, and recognizing objects in 2D / 3D more accurately than methods based on a single image. We see this work as a promising step toward the goal of coherently interpreting the geometrical and semantic content of complex scenes.

**Acknowledgment.** We acknowledge the support of NSF CAREER #1054127 and the Gigascale Systems Research Center. We wish to thank Mohit Bagra for his help in collecting the Kinect dataset and Min Sun for helpful feedback.

## 7 Appendix

**Proposition 1.** *Assume that at least 3 objects can be detected in the  $k^{\text{th}}$  image. Assume that the detector returns object image coordinates  $x_t^k, y_t^k$  ( $t = 1, 2, 3$ ), scales  $w_t^k, h_t^k$ , and category  $c_t$ . Assume that the mappings  $W_t$  and  $H_t$  are available for each detected object. Then extrinsic camera parameters  $R^k, T^k$  can be calculated.*

*Proof.* We demonstrate proposition 1 for 3 objects but the proof can be extended if more than 3 objects are available. Let  $O_1, O_2, O_3$  be the observed objects and  $O_1^k, O_2^k, O_3^k$  are their locations, poses, scales in the  $k^{\text{th}}$  camera reference system. We define the world reference system based on the first camera: location of  $O_1^1$  is the origin; the vector from  $O_2^1$  to  $O_1^1$  is the X-axis; and the locations of  $O_1^1, O_2^1, O_3^1$  (3 points) characterize the X-Y plane. The object coordinate in camera reference system is  $[X_t^k, Y_t^k, Z_t^k] = Z_t^k (K^k)^{-1} [x_t, y_t, 1]'$ , where  $Z_t^k$  can be computed from  $w_t^k, h_t^k$  with the mappings  $W$  and  $H$ . Therefore, we have the camera translation as  $T^k = [X_1^k, Y_1^k, Z_1^k]$ . Since  $[x_t, y_t, 1]' = K^k (R^k [X_t, Y_t, Z_t]' + T_k) / Z_t^k$  ( $t = 1, 2, 3$ ) and the degree of freedom of  $R^k$  is 3, the camera rotation matrix  $R^k$  can be solved accordingly.

**Proposition 2.** *Assume that at least 2 objects can be detected in all the images. Assume that from image  $k$  the detector returns object image coordinates  $x_t^k, y_t^k$ , pose  $\theta_t^k, \phi_t^k$ , and category  $c_t$ . The camera extrinsic parameters  $R^k, T^k$  can be calculated up to a scale ambiguity (metric reconstruction).*

*Proof.* We demonstrate proposition 2 for 2 objects but the proof can be extended if more than 2 objects are available. Let  $O_1, O_2$  be the observed objects, and let  $O_1^k, O_2^k$  be their locations, poses, scales in the  $k^{\text{th}}$  camera reference system. We define the world reference system based on the first camera: the location  $O_1^1$  is the origin; and the normals (q,t,n) of the bounding cube of  $O_1^1$  (Figure 2) are the X,Y,Z axes. To address the ambiguity of the metric construction, we assume the distance between  $O_1$  and  $O_2$  is unit length. By using the observed pose of  $O_1^k$  and  $O_1^1$ , the rotation of the  $k^{\text{th}}$  camera  $R^k$  can be computed, and its translation  $T^k$  is unknown up to 1 degree of freedom which is the distance of  $C^k$  to  $O_1$ . Since we assume the distance between  $O_1, O_2$  is unit length, the 3D location of  $O_2$  (in the world system) becomes a function of  $T^k$ , denote which by  $X_2(T^k), Y_2(T^k), Z_2(T^k)$ . Given all the cameras  $C^1 \dots C^{N_k}$ , we have equations  $[X_2(T^1), Y_2(T^1), Z_2(T^1)] = [X_2(T^2), Y_2(T^2), Z_2(T^2)] = \dots = [X_2(T^{N_k}), Y_2(T^{N_k}), Z_2(T^{N_k})]$ . These equations provide  $3 \times (N_k - 1)$  constraints. Since the degree of freedom of  $T^k$  is 1, the number of unknowns are  $N_k$ . Therefore  $\{T^k\}$  can be jointly solved if more than two cameras are available. Notice that the  $\{R^k, T^k\}$  are estimated by assuming  $O_1, O_2$  has the unit length. However,

the real distance of  $O_1, O_2$  is unknown and therefore the estimation of cameras is up to a metric reconstruction.

**Proposition 3.** *Assume that at least 1 object can be detected in all the images. Assume on image  $k$  the detector returns object image coordinates  $x_t^k, y_t^k$ , pose  $\theta_t^k, \phi_t^k$ , scales  $w_t^k, h_t^k$ , and category  $c_t$ . Assume that the mapping  $W_t$  and  $H_t$  are available for each detected object. Then the camera extrinsic parameters  $R^k, T^k$  can be calculated.*

*Proof.* We demonstrate proposition 3 for 1 object but the proof can be extended if more than 1 object is available. Let  $O_1$  be the observed object and  $O_1^k$  be its location, pose, and scale in the  $k^{th}$  camera reference system. We define the world reference system based on the first camera: the location of  $O_1^1$  is the origin and the normals (q,t,n) of the 3D cube of  $O_1$  (Figure 2) are the X,Y,Z axes. Hence,  $\Theta_1, \Phi_1$  (in the world system) is the same as the observed  $\Theta_1^1, \Phi_1^1$ . Object camera coordinate is  $[X_1^k, Y_1^k, Z_1^k] = Z_1^k (K^k)^{-1} [x_1, y_1, 1]'$ . Therefore, the translation of the  $k^{th}$  camera is  $T^k = [X_1^k, Y_1^k, Z_1^k]$ . Finally,  $R^k$  can be computed by  $\theta_1^k, \phi_1^k$  and  $\Theta_1, \Phi_1$ .

## References

1. Bao, S.Y., Savarese, S.: Semantic structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (2011)
2. Bao, S.Y., Sun, M., Savarese, S.: Toward coherent object detection and scene layout understanding. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (2010)
3. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and Recognition Using Structure from Motion Point Clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
4. Cheng, Y.: Mean shift, mode seeking, and clustering. PAMI (1995)
5. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.: 3d urban scene modeling integrating recognition and reconstruction. IJCV 78(2-3), 121–141 (2008)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (2005)
7. Dellaert, F., Seitz, S., Thrun, S., Thorpe, C.: Feature correspondence: A markov chain monte carlo approach. In: NIPS (2000)
8. Dick, A.R., Torr, P.H.S., Cipolla, R.: Modelling and interpretation of architecture from several images. IJCV 60(2), 111–134 (2004)
9. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. TPAMI (2009)
10. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR, vol. 2, pp. 264–271 (2003)
11. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing Objects in Range Data Using Regional Point Descriptors. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
12. Gilks, W., Richardson, S., Spiegelhalter, D.: Markov chain Monte Carlo in practice. Chapman and Hall (1996)

13. Golparvar-Fard, M., Pena-Mora, F., Savarese, S.: D4ar- a 4-dimensional augmented reality model for automating construction progress data collection, processing and communication. In: TCON Special Issue: Next Generation Construction IT (2009)
14. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
15. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press (2000)
16. Helmer, S., Meger, D., Muja, M., Little, J., Lowe, D.: Multiple viewpoint recognition and localization. In: ACCV (2011)
17. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. *International Journal of Computer Vision* 80(1) (2008)
18. Huber, D.: Automatic 3d modeling using range images obtained from unknown viewpoints. In: *Int. Conf. on 3-D Digital Imaging and Modeling* (2001)
19. Khan, S.M., Shah, M.: A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3954, pp. 133–146. Springer, Heidelberg (2006)
20. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2006)
21. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: *ECCV 2004 Workshop on Statistical Learning in Computer Vision* (2004)
22. Li, L.-J., Socher, R., Fei-Fei, L.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: *CVPR* (2009)
23. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* (2004)
24. Nister, D.: An efficient solution to the five-point relative pose problem. *TPAMI* (2004)
25. Pandey, G., McBride, J.R., Eustice, R.M.: Ford campus vision and lidar data set. *International Journal of Robotics Research* (2011)
26. Pollefeys, M., Gool, L.V.: From images to 3d models. *Commun. ACM* 45(7), 50–55 (2002)
27. Reynolds, M., Doboš, J., Peel, L., Weyrich, T., Brostow, G.J.: Capturing time-of-flight data with confidence. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2011)
28. Rusu, R., Marton, Z., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems* 56(11) (2008)
29. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: *ICCV* (2007)
30. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. *PAMI* 31(5), 824–840 (2009)
31. Snavely, N., Seitz, S.M., Szeliski, R.S.: Modeling the world from internet photo collections. *IJCV* (2) (2008)
32. Soatto, S., Perona, P.: Reducing "structure from motion": a general framework for dynamic vision. part 1: modeling. *International Journal of Computer Vision* 20 (1998)
33. Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Depth from familiar objects: A hierarchical model for 3d scenes. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2006)
34. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: a modern synthesis. In: *Vision Algorithms: Theory and Practice* (1999)
35. Tuytelaars, T., Van Gool, L.: Wide baseline stereo matching based on local, affinity invariant regions. In: *British Machine Vision Conference* (2000)

# Hierarchical Surface Reconstruction from Multi-resolution Point Samples

Ronny Klowsky, Patrick Mücke, and Michael Goesele

TU Darmstadt

**Abstract.** Robust surface reconstruction from sample points is a challenging problem, especially for real-world input data. We present a new hierarchical surface reconstruction based on volumetric graph-cuts that incorporates significant improvements over existing methods. One key aspect of our method is, that we exploit the footprint information which is inherent to each sample point and describes the underlying surface region represented by that sample. We interpret each sample as a vote for a region in space where the size of the region depends on the footprint size. In our method, sample points with large footprints do not destroy the fine detail captured by sample points with small footprints. The footprints also steer the inhomogeneous volumetric resolution used locally in order to capture fine detail even in large-scale scenes. Similar to other methods our algorithm initially creates a crust around the unknown surface. We propose a crust computation capable of handling data from objects that were only partially sampled, a common case for data generated by multi-view stereo algorithms. Finally, we show the effectiveness of our method on challenging outdoor data sets with samples spanning orders of magnitude in scale.

## 1 Introduction

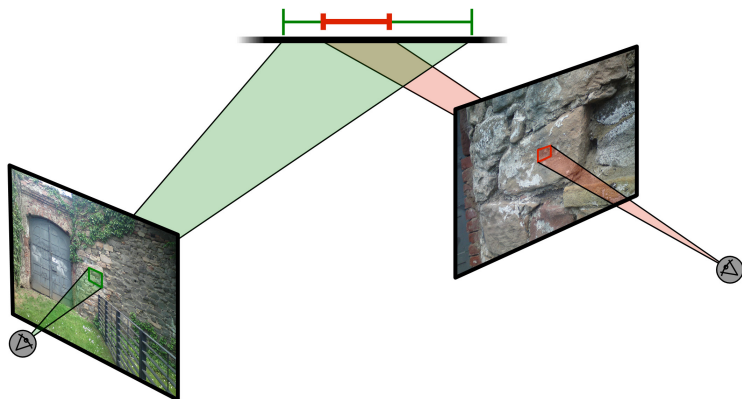
Reconstructing a surface mesh from sample points is a problem that occurs in many applications, including surface reconstruction from images as well as scene capture with triangulation or time-of-flight scanners. Our work is motivated by the growing capabilities of multi-view stereo (MVS) techniques [20,8,9,7] that achieve remarkable results on various data sets.

Traditionally, surface reconstruction techniques are designed for fairly high-quality input data. Measured sample points, in particular samples generated by MVS algorithms, are, however, *noisy* and contain *outliers*. Figure 1 shows an example reconstructed depth map that we use as input data in our method. Furthermore, sample points are often non-uniformly distributed over the surface and entire regions might not be represented at all. Recently, Hornung and Kobbelt presented a robust method well suited for noisy data [12]. This method generates optimal low-genus watertight surfaces within a crust around the object using a volumetric graph cut. Still, their algorithm has some major limitations regarding crust generation, sample footprint, and missing multi-resolution reconstruction which we address in this paper.





**Fig. 1.** *Left:* An input image to Multi-View Stereo reconstruction. *Middle:* The reconstructed depth map visualized in gray values (white: far, black: near). *Right:* The triangulated depth map rendered from a slightly different view point.



**Fig. 2.** Visualization of the *footprint* of a sample point: A certain pixel in the left image covers a significantly larger area than a corresponding pixel in the right image

Hornung and Kobbelt create a surface confidence function based on unsigned distance values extracted from the sample points. The final surface  $S$  is obtained by optimizing for maximum confidence and minimal surface area. As in many surface reconstruction algorithms, the footprint of a sample point is completely ignored when computing the confidence. Every sample point, regardless of how it was obtained, inherently has a *footprint*, the underlying surface area taken into account during the measurement (see Figure 2). The size of the footprint indicates the sample point’s capability to capture surface details. A method that outputs sample points with different footprints was proposed by Habbeke and Kobbelt [9]. They represent the surface with surfels (surface elements) of varying size depending on the image texture. Furukawa et al. [7] consider footprints to estimate reconstruction accuracy and Fuhrmann and Gesele [6] build a hierarchical signed distance field where they insert samples on different scales depending on their footprint. However, both methods effectively discard samples with large footprints prior to final surface extraction. In this paper, we propose a different way to model the sample footprint during the reconstruction process. In particular, we create a modified confidence map where samples contribute differently depending on their footprints.

The confidence map is only evaluated inside a *crust*, a volumetric region around the sample points. In [12], the crust computation implicitly segments the boundary of the crust into *interior* and *exterior*. The final surface separates interior from exterior. This crust computation basically works only for completely sampled objects. Even with their proposed workaround (estimating the medial axis), the resulting crust is still not applicable to many data sets. Such a case is illustrated in Figure 3, where no proper interior component can be computed. This severely restricts the applicability of the entire algorithm. We propose a different crust computation that separates the crust generation from the crust segmentation process, extending the applicability to a very general class of input data.

Finally, as Vu et al. [24] pointed out, volumetric methods such as [12] relying on regular volume decomposition are not able to handle large-scale scenes. To overcome this problem our algorithm reconstructs on a locally adaptive volumetric resolution and finally extracts a watertight surface. This allows us to reconstruct fine details even in large-scale scenes such as the Citywall data set (see Figure 11).

This paper builds strongly on a recent publication by Mücke et al. [18] but contains the following substantial improvements.

- The sampling of the global confidence map is parallelized.
- We now employ a graph embedding modeling the 26-neighborhood which better approximates the Euclidean distance.
- Surface extraction is deferred to the end of the algorithm by using a combination of marching cubes and marching tetrahedra on a multi-resolution grid. This supersedes the need of the error-prone mesh clipping used before.

In addition, we show the effectiveness of our algorithm on a new challenging data set with high surface genus.

The remainder of the paper is organized as follows: First, we review previous work (Section 2) and give an overview of our reconstruction pipeline (Section 3). Details of the individual steps are explained in Sections 4–7. Finally, we present results of our method on standard benchmark data as well as challenging outdoor scenes (Section 8) and wrap up with a conclusion and an outlook on future work (Section 9).

## 2 Related Work

### Surface Reconstruction from (Unorganized) Points

Surface reconstruction from unorganized points is a large and active research area. One of the earliest methods was proposed by Hoppe et al. [10]. Given a set of sample points, they estimate local tangent planes and create a signed distance field. The zero-level set of this signed distance field, which is guaranteed to be a manifold, is extracted using a variant of the marching cubes algorithm [15].

If the sample points originate from multiple range scans, additional information is available. VRIP [5] uses the connectivity between neighboring samples as

well as the direction to the sensor when creating the signed distance field. Additionally, it employs a cumulative weighted signed distance function allowing it to incrementally add more data. The final surface is again the zero-level set of the signed distance field. A general problem of signed distance fields is that local inconsistencies of the data lead to surfaces with undesirably high genus and topological artifacts. Zach et al. [25] mitigate this effect. They first create a signed distance field for each range image and then compute a regularized field  $u$  approximating all input fields while minimizing the total variation of  $u$ . The final surface is the zero-level set of  $u$ . Their results are of good quality, but the resolution of both, the volume and the input images, is very limited. In their very recent paper, Fuhrmann and Goesele [6] introduce a depth map fusion algorithm that takes sample footprints into account. They merge triangulated depth maps into a hierarchical signed distance field similar to VRIP. After a regularization step, basically pruning low-resolution data where reliable higher-resolution data is available, the final surface is extracted using marching tetrahedra. Our method does not rely on triangulated depth maps and tries to merge all data samples while never discarding information from low-resolution samples. Another recent work taking unorganized points as input is called cone carving and is presented by Shalom et al. [21]. They associate each point with a cone around the estimated normal to carve free space and obtain a better approximation of the signed distance field. This method is in a way characteristic for many surface reconstruction algorithms in the sense that it is designed to work on raw scans from a commercial 3D laser scanner with rather good quality. Such methods are often not able to deal with the lower quality data generated by MVS methods from outdoor scenes containing a significant amount of noise and outliers.

Kazhdan et al. [13] reformulate the surface reconstruction problem as a standard Poisson problem. They reconstruct an indicator function marking regions inside and outside the object. Oriented points are interpreted as samples of the gradient of the indicator function, requiring accurate normals at each sample point's position which are usually not present in MVS data. The divergence of the smoothed vector field, represented by these oriented points, equals the Laplacian of the indicator function. The final surface is extracted as an iso-surface of the indicator function using a variant of the marching cubes algorithm. Along these lines, Alliez et al. [1] use the normals to derive a tensor field and compute an implicit function whose gradients best approximate that tensor field. Additionally, they present a technique, called Voronoi-PCA, to estimate unoriented normals using the Voronoi diagram of the point set.

## Graph Cut Based Surface Reconstruction

Boykov and Kolmogorov [2] introduced the idea of reconstructing surfaces by computing a cut on a graph embedded in continuous space. They also show how to build a graph and set the edge weights such that the resulting surface is minimal for any anisotropic Riemannian metric. Hornung and Kobbelt [11] use the volumetric graph cut to reconstruct a surface given a photo-consistency measure defined at each point of a predefined volume space. They propose to

embed an octahedral graph structure into the volume and show how to extract a mesh from the set of cut edges. In a follow-up paper [12], they present a way to compute confidence values from a non-uniformly sampled point cloud and improve the mesh extraction procedure.

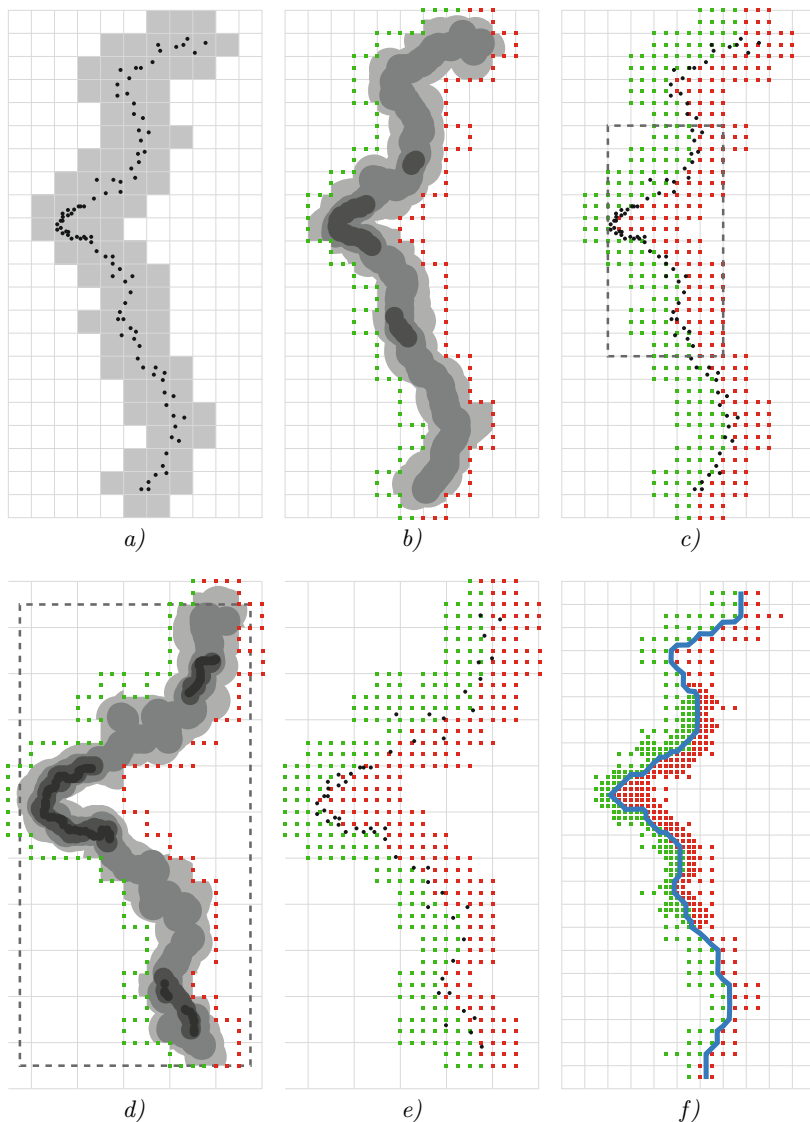
An example of using graph cuts in multi-view stereo is the work of Sinha et al. [22]. They build an adaptive multi-resolution tetrahedral mesh where an estimated photo-consistency guides the subdivision. The final graph cut is performed on the dual of the tetrahedral mesh followed by a photo-consistency driven mesh refinement. Labatut et al. [14] build a tetrahedral mesh around points merged from multiple range images. They introduce a surface quality term and a surface visibility term that takes the direction to the sensor into account. From an optimal cut, which minimizes the sum of the two terms, a labeling of each tetrahedra as inside or outside can be inferred. The final mesh consists of the set of triangles separating the tetrahedra according to their labels. Vu et al. [24] replace the point cloud obtained from multiple range images with a set of 3D features extracted from the images. The mesh obtained from the tetrahedral graph cut is refined mixing photo-consistency in the images and a regularization force. However, none of the existing graph cut based surface reconstruction algorithms properly incorporates the footprint of a sample.

### 3 Overview

The input of our algorithm is a set of *surface samples* representing the scene (Figure 3a). Each surface sample consists of its position, footprint size, a scene surface normal approximation, and an optional confidence value. A cubic bounding box is computed from the input points or given by the user.

First, we determine the *crust*, a subset of the bounding volume containing the unknown surface. All subsequent computations will be performed inside this crust only. Furthermore, the boundary of the crust is partitioned into *interior* and *exterior*, defining interior and exterior of the scene (Figure 3b). Inside the crust we compute a *global confidence map*, such that points with high confidence values are likely to lie on the unknown surface. Each sample point adds confidence to a certain region of the volume. The size of the region and the confidence peak depend on the sample point’s footprint size. Effectively, every sample point adds the same total amount of confidence to the volume but spread out differently. A volumetric graph is embedded inside the crust where graph nodes correspond to voxels and graph edges map the 26-neighborhood. A minimal cut on this graph separates the voxels into interior and exterior representing the optimal surface at this voxel resolution (Figure 3c). The edge weights of the graph are chosen such that the final surface minimizes surface area while maximizing confidence.

We then identify surface regions with sampled details too fine to be adequately represented on the current resolution. Only these regions are subdivided, the global confidence map is resampled, and the graph cut is computed on a higher resolution (Figure 3d+e). We repeat this process iteratively until eventually all fine details were captured. Finally, we extract the surface in the irregular



**Fig. 3.** Overview of our reconstruction pipeline. *a)* We compute a crust around the input samples of different footprints and varying sampling density. *b)* We segment the crust into *interior* (red) and *exterior* (green) and compute the global confidence map (GCM) to which each input sample contributes. *c)* A minimal cut on the embedded graph segments the voxel corners representing the surface with maximum confidence while minimizing surface area. We mark the areas with high-resolution samples (dashed black box) and iteratively increase resolution therein. *d+e)* In the increased resolution area we re-evaluate the GCM and perform the graph cut optimization. *f)* Finally, an adaptive triangle mesh is extracted from the multi-resolution voxel corner labeling.

voxel grid using a combination of marching cubes and marching tetrahedra. This results in a multi-resolution surface representation of the scene, the output of our algorithm (Figure 3f).

## 4 Crust Computation

We subdivide the cubic bounding box into a regular voxel grid. For memory efficiency and to easily increase the voxel resolution, this voxel grid is represented by an octree data structure. Our algorithm iteratively treats increasing octree levels (finer resolution) starting with a user-defined low octree level  $\ell_0$ , i.e., with a coarse resolution.

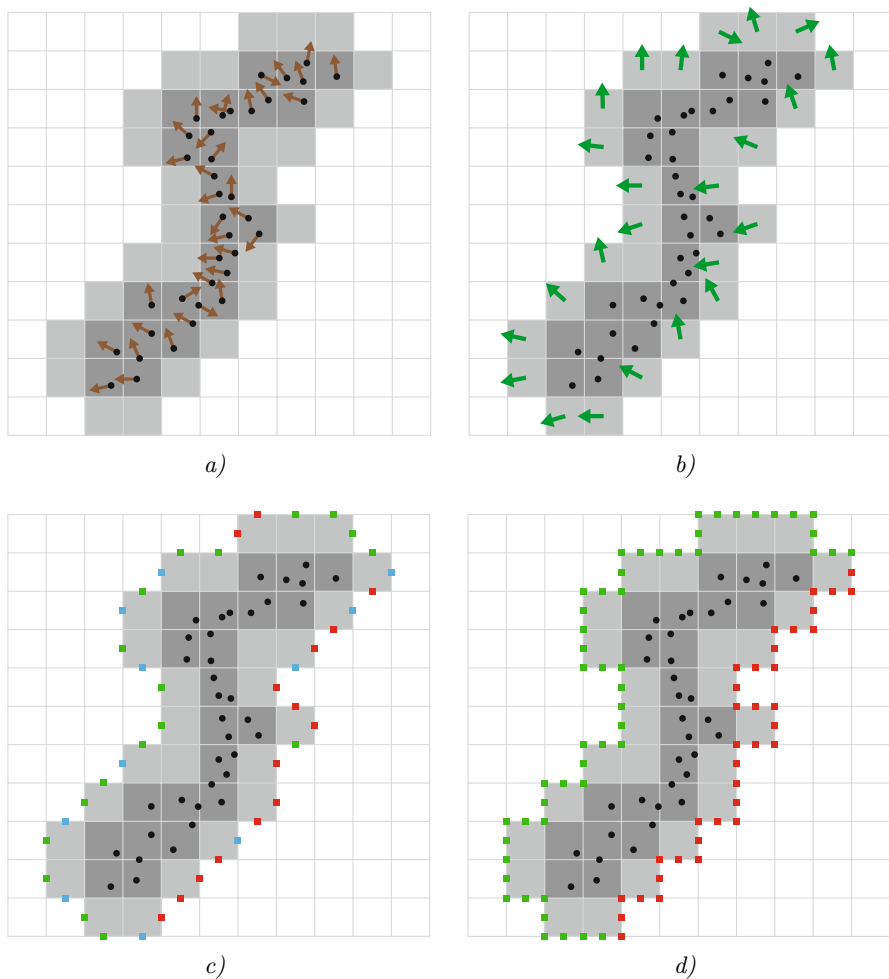
The crust  $V_{crust} \subset V$  is a subset of voxels that contains the unknown surface. The crust computation is an important step in the algorithm for several reasons: The shape of the crust constrains the shape of the reconstructed surface. Furthermore, the crust has to be sufficiently large to contain the optimal surface and on the other hand as narrow as possible to reduce computation time and memory cost. We split the crust computation into two parts. First, the crust is generated, then the boundary of this crust is segmented to define interior and exterior of the scene (see Figure 4 for an overview).

*Crust Generation.* We initialize the crust on level  $\ell_0$  with the set of voxels on the parent octree level  $\ell_0 - 1$  containing surface samples. We dilate this sparse set of voxels several times over the 6-neighborhood of voxels, followed by a morphological closing operation (Figure 4a). The number of dilation steps is currently set by the user, but the resulting crust shape can be immediately inspected, as the crust generation is fast on the low initial resolution. Subsequently, these voxels  $v \in V_{crust}^{\ell_0-1}$  are once regularly subdivided to obtain the initial crust  $V_{crust}^{\ell_0}$  for further computations on level  $\ell_0$ .

*Crust Segmentation.* In this step our goal is to assign labels *interior* and *exterior* to all boundary voxel corners on level  $\ell_0$  to define the interior and exterior of the scene. In the following, we define  $\partial V_{crust}^\ell$  to be the set of boundary voxels on level  $\ell$ . We start by determining labels for voxel corners  $v_f$  that lie on the midpoints of boundary faces of parent crust voxels  $v \in \partial V_{crust}^{\ell_0-1}$ . The labels are determined by comparing a surface normal estimate  $\mathbf{n}_v^{surf}$  for parent voxel  $v$  with the normals of the boundary faces  $\mathbf{n}_{v_f}^{crust}$ . The surface normal is computed for each crust voxel by averaging the normals of all sample points inside the crust voxel. Crust voxels that do not contain surface samples obtain their normal estimate through propagation during crust dilatation (Figure 4b). We determine the initial labels on the crust boundary by

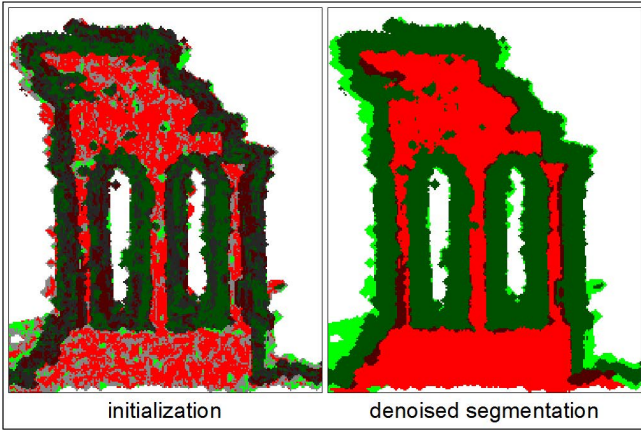
$$label(v_f) = \begin{cases} exterior, & \text{if } \mathbf{n}_{v_f}^{crust} \cdot \mathbf{n}_v^{surf} \geq \tau \\ interior, & \text{if } \mathbf{n}_{v_f}^{crust} \cdot \mathbf{n}_v^{surf} \leq -\tau \\ unknown, & \text{otherwise} \end{cases} \quad (1)$$

with  $\tau \in (0, 1)$  (Figure 4c). We used  $\tau = 0.75$  in all experiments.



**Fig. 4.** Initial crust computation for lowest resolution: *a)* We initialize the crust with voxels containing sample points and dilate several times. *b)* Surface normals are computed for each voxel. *c)* The comparison of surface normals with the face normals of the crust voxels defines an initial labeling into *interior* (red), *exterior* (green), and *unknown* (blue). *d)* An optimization yields a homogenous crust surface segmentation.

By now we have just labeled a subset of all voxel corners on level  $\ell_0$  (Figure 4c). Furthermore, since surface normal information of the samples may only be a crude approximation, this initial labeling is noisy and has to be regularized. We cast the problem of obtaining a homogenous labeling of the crust surface into a 2D binary image denoising problem solved using graph cut optimization as described by Boykov and Veksler [4]. We build a graph with a node per voxel corner in  $\partial V_{crust}^{\ell_0}$  and a graph edge connecting two nodes if the corresponding voxel corners share a voxel edge. Additionally, ‘diagonal’ edges are inserted that



**Fig. 5.** Visualization of the crust surface for the Temple (cut off perpendicular to the viewing direction). The color is similar to Figure 4. Light shaded surfaces are seen from the front, dark shaded ones are seen from the back.

connect the initially labeled corners in the middle of parent voxel faces with the four parent voxel corners. We also add two terminal nodes *source* and *sink* together with further graph edges connecting each node to these terminals. Note that this graph is used for the segmentation of the crust on the lowest resolution level  $\ell_0$  only and should not be confused with the graphs used for surface reconstruction on the different resolutions.

All edges connecting two non-terminal nodes receive the same edge weight  $w$ . Edges connecting a node  $n$  with a terminal node receive a weight depending on the labeling of the corresponding voxel corner  $v_c$ , where unlabeled voxel corners are treated as unknown:

$$w_n^{source} = \begin{cases} \mu & \text{if } v_c \text{ is labeled } interior \\ 1 - \mu & \text{if } v_c \text{ is labeled } exterior \\ \frac{1}{2} & \text{if } v_c \text{ is unknown} \end{cases} \quad (2)$$

$$w_n^{sink} = 1 - w_n^{source} \quad (3)$$

for a constant  $\mu \in (0, \frac{1}{2})$ . With these edge weights the *exterior* is associated with *source*, *interior* with *sink*. A cut on this graph assigns each node either to the *source* or to the *sink* component and therefore yields a homogeneous segmentation of the boundary voxel corners of  $\partial V_{crust}^{\ell_0}$  (Figure 4d and Figure 5 right). We used  $w = 0.5$  and  $\mu = 0.25$  in all experiments.

If two neighboring crust voxel corners obtained different labels, the reconstructed surface is forced to pass between them, as it has to separate *interior* from *exterior*. The denoising minimizes the number of such occurrences and therefore prevents unwanted surfaces from being formed. In the case of entirely sampled surfaces and a correctly computed crust, two neighboring voxel corners never have different labels. However, if the scene surface is not sampled entirely,



such segment borders occur even for correct segmentations (see Figure 4d). This forces the surface to pass through the two involved voxel corners which, unlike the rest of the surface reconstruction, does not depend on the confidence values. This fixation does not affect the surface in sampled regions, though. We exploit this constraint on the reconstructed surface in our refinement step where we reconstruct particular areas on higher resolution (see Section 7).

## 5 Global Confidence Map

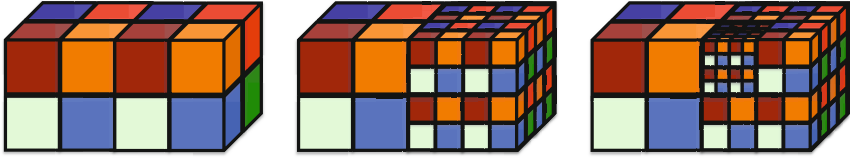
The *global confidence map* (GCM) is a mapping  $\Gamma : \mathbb{R}^3 \rightarrow \mathbb{R}$  that assigns a confidence value to each point in the volume. Our intuition is that each sample point spreads its confidence over a region in space whose extent depends on the sample footprint. Thus, sample points with a small footprint create a focused spot whereas sample points with a large footprint create a blurry blob (see Figure 3b). We model the spatial uncertainty of a sample point as a Gaussian  $\gamma_s$  centered at the sample point’s position with standard deviation equal to half the footprint size. If the sample points are associated with confidence values we scale the Gaussian accordingly. The *local confidence map* (LCM)  $\gamma_s$  determines the amount of confidence added by a particular sample point  $s$ . Consequently, the GCM is the sum over all LCMs:

$$\Gamma(x) = \sum_s \gamma_s(x). \quad (4)$$

*Implementation.* Let  $\ell$  be the octree level at which we want to compute the graph cut. In all crust voxels  $\{x_v\}_{v \in V_{crust}^\ell}$  we evaluate the GCM  $\Gamma$  at 27 positions: at the 8 corners of the voxel, at the middle of each face and edge, and at the center of the voxel. When adding up the LCMs of each sample point  $s$  we clamp the value of  $\gamma_s$  to zero for points for which the distance to  $s$  is larger than three times the footprint size of sample point  $s$ . Also, we sample each  $\gamma_s$  only at a fixed number of positions ( $\approx 5^3$ ) within its spatial support and exploit the octree data structure by accumulating each  $\gamma_s$  to nodes at the appropriate octree level depending on the footprint size. After all samples have been processed, the accumulated values in the octree are propagated to the nodes at level  $\ell$  by adding the values at a node to the children’s nodes using linear interpolation for in-between positions. The support of LCMs of sample points with small footprints might be too narrow to be adequately sampled on octree level  $\ell$ . For those samples we temporarily increase the footprint for the computation of the LCM  $\gamma_s$  and mark the corresponding voxel for later processing at higher resolution.

### 5.1 Parallelization

In order to speed-up the sample insertion into the octree which is costly since each input point creates  $\approx 125$  samples, we parallelize the insertion at each octree level  $\hat{\ell} \leq \ell$  using a binning approach. In our implementation, bins correspond to voxels. In each bin we sort the samples into eight lists representing the eight child



**Fig. 6.** Visualization of an intermediate state of the binning approach used for the parallelization of the GCM computation. Starting with two bins (left), the right bin is subdivided into eight new bins (middle). One of the new bins is subdivided again (right) resulting in a total number of 16 bins.

voxels in a predefined order. We process the first list of all bins in parallel, then the second list, and so on. For this purpose samples in list  $x$  of two different bins should not interfere with each other, i.e., affect the same nodes in the octree. We start with the bounding cube as root bin containing all samples to be processed on level  $\hat{\ell}$ . We subdivide a bin if the following two criteria are satisfied:

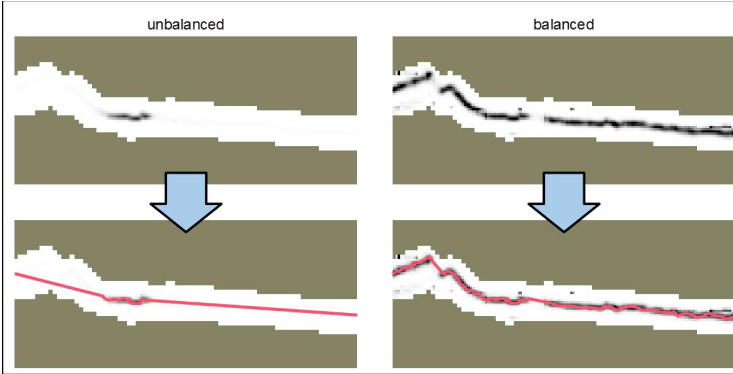
1. the bin contains more than  $n_{max}$  samples, and
2. subdividing the bin maintains the property that samples out of the same list but different bins do not interfere with each other given their footprint.

When subdividing a bin the lists are effectively turned into bins and the samples are partitioned into eight smaller lists according to the same predefined order as before. The subdivision stops if a maximum number of bins has been reached or no more bins can be subdivided. Figure 6 shows the main principle of the subdivision process where the color coded voxels represent the individual lists. Note that two voxels with the same color never touch so that the LCM of samples do not interfere with each other.

## 6 Graph Cut

As done by Hornung and Kobbelt [12] we apply a graph cut to find the optimal surface. The layout of the graph cut is however more similar to Boykov and Kolmogorov [2] since we define a graph node per voxel and edges representing the 26-neighborhood (inside the set of crust voxels  $V_{crust}$ ). Note that at this stage we compute the graph cut on a certain resolution only and do not extract the surface explicitly. The edge weights  $w_i$  in the graph are derived from the GCM values  $\Gamma(x_i)$  in the center of the voxel, edge, or face, respectively. Since the optimal surface should maximize the global confidence  $\Gamma$  we want to set small edge weights for regions with high confidence and vice versa. A straightforward way to implement this would be

$$w_i = 1 - \frac{\Gamma(x_i)}{\Gamma_{max}} + a \quad \text{with} \quad \Gamma_{max} = \max_{x \in \mathbb{R}^3} \Gamma(x) \quad (5)$$



**Fig. 7.** The GCM values can be arbitrarily large leading to near-constant edge weights in large regions of the volume (left). Our *local GCM balancing* compensates for that allowing the final graph cut to find the correct surface (right).

such that all edge weights lie in  $[a, 1 + a]$ , where  $a$  controls the surface tension. Note, that scaling all edge weights with a constant factor does not change the resulting set of cut edges. As the global maximum  $\Gamma_{max}$  can be arbitrarily large, local fluctuation of the GCM might be vanishingly small in relation to  $\Gamma_{max}$  (see Figure 7 left). Since the graph cut also minimizes the surface area while maximizing for confidence, the edge weights need to have sufficient local variation to avoid that the graph cut only minimizes the number of cut edges and thus the surface area (*shrinking bias*). In order to cope with that, we apply a technique similar to an adaptive histogram equalization which we call *local GCM balancing*. Instead of using the global maximum in Equation 5 we replace it with the weighted local maximum (LM) of the GCM at point  $x$ . We compute  $\Gamma_{LM}(x)$  by

$$\Gamma_{LM}(x) = \max_{y \in \mathbb{R}^3} \left[ W \left( \frac{\|x - y\|}{2^{-\ell} \cdot \mathcal{B}_{edge}} \right) \cdot \Gamma(y) \right] \quad (6)$$

where  $\mathcal{B}_{edge}$  is the edge length of the bounding cube. We employ a weighting function  $W$  to define the scope in which the maximum is computed. We define  $W$  as

$$W(d) = \begin{cases} 1 - \left( \frac{d}{\frac{1}{2}\mathcal{D}} \right)^c & \text{if } d \leq \frac{1}{2}\mathcal{D} \\ 0 & \text{if } d > \frac{1}{2}\mathcal{D} \end{cases} \quad (7)$$

where  $\mathcal{D}$  is the filter diameter in voxels. We used  $\mathcal{D} = 11$  and  $c = 4$  in all our experiments.  $W$  is continuous in order to ensure continuity of the GCM. See Figure 7 (right) to see the effect of local GCM balancing.

After the graph cut, each voxel corner on octree level  $\ell$  is either labeled interior or exterior which we can think of as binary signed distance values. In particular,

since the subdivision from level  $\ell - 1$  is regular we have labels for all voxel corners, the voxel center, the center of each face and edge. This will be exploited during final surface extraction in the next Section.

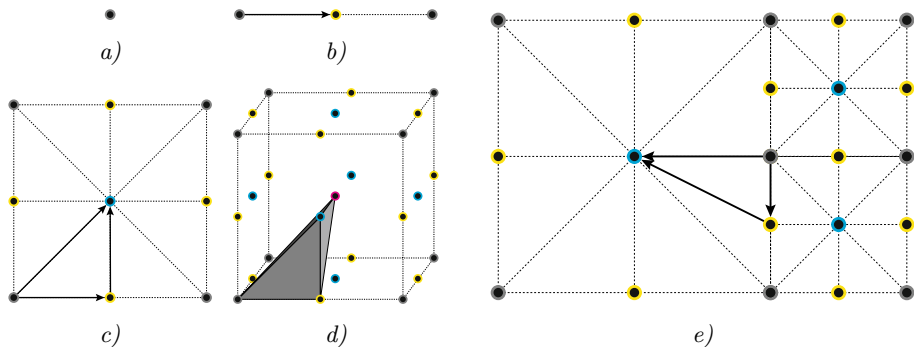
## 7 Multi-resolution Surface Reconstruction

Due to memory limitations, it is often impossible to reconstruct the whole scene on a resolution high enough to capture all sampled details. An adaptive multi-resolution approach which reconstructs different scene regions on adaptive resolutions depending on the sample footprints is therefore desirable. During the GCM sampling on octree level  $\ell$  we marked voxels that need to be processed on higher resolution. After the graph cut we dilate this set of voxels several times and regularly subdivide the resulting voxel set to obtain a new crust  $V_{crust}^{\ell+1}$ . The crust segmentation can be obtained from the graph cut on level  $\ell$ , as this cut effectively assigns each voxel corner a label *interior* or *exterior*. For boundary voxel corners in  $V_{crust}^{\ell+1}$  that coincide with voxel corners on level  $\ell$  we simply transfer the label. This ensures a continuous reconstruction across level boundaries. For voxel corners that lie on a parent voxel edge or face, i.e., between two or four voxel corners on level  $\ell$ , we obtain the conform label of the surrounding voxel corners or we leave it unknown. The new crust  $V_{crust}^{\ell+1}$  is now ready for graph cut optimization on level  $\ell + 1$  (see Figure 3d+e). For voxel corners that coincide with voxel corners on the lower resolution the resulting labeling on level  $\ell + 1$  overwrites the labeling obtained before.

The recursive refinement stops if the maximum level  $\ell_{max}$  is reached or no voxels are marked for further processing. Due to our refinement scheme the last subdivision in the octree is always regular, i.e., all eight octants are present. The graph cuts define the voxel corners of the finest voxels as interior or exterior.

### 7.1 Final Surface Extraction

To extract the final surface we apply a combination of marching cubes and marching tetrahedra. The decision is made voxel-by-voxel one level above the finest level. Note that the last subdivision step is always regular. If the voxel is single-resolution containing 27 labeled voxel corners, we apply classical marching cubes to all eight child voxels. We interpret the voxel corner labels as binary signed distance values. If the voxel is multi-resolution, i.e., there is a change in resolution present affecting at least one of the cube edges or faces, we apply the tetrahedralization scheme by Manson and Schaefer [16] (see Figure 8). We hereby place dual vertices at voxel corners and at the centers of edges, faces, and voxels. These positions coincide with voxel corners of the finest levels providing the binary signed distance values needed for the subsequent marching tetrahedra. Now, we only need to take care of voxel faces where triangles produced by marching cubes and triangles produced by marching tetrahedra meet. It is possible that T-vertices were created here but this can be easily fixed using



**Fig. 8.** Tetrahedralization of the multi-resolution grid. We connect a vertex (a) with the dual vertex of an edge (b), add a face vertex (c), and form a tetrahedron by adding the dual vertex of a cell (d). Adaptive triangulation of the multi-resolution grid (e). Tetrahedralization scheme and figures similar to Manson and Schaefer [16].

an edge flip or vertex collapse. The final multi-resolution surface mesh is watertight and has different sized triangles depending on the details present in the corresponding areas.

## 8 Results

We will now present results of our method on different data sets (see Table 1). The source code is publicly available on the project page [19]. Our experiments were performed on a 2.7 GHz AMD Opteron with eight quad-core processors and 256GB RAM. All input data was generated from images using a robust structure-from-motion system [23] and an implementation of a recent MVS algorithm [8] applied to down-scaled images. We used all reconstructed points from all depth maps as input samples for our method. The footprint size of a sample is computed as the diameter of a sphere around the sample’s 3D position whose projected diameter in the image equals the pixel spacing. For all graph cuts involved we used the publicly available library by Boykov and Kolmogorov [3].

**Table 1.** The data sets we used and the number of sample points, the number of vertices in the resulting meshes, octree levels used for surface extraction, computation time and relative variation in footprint size

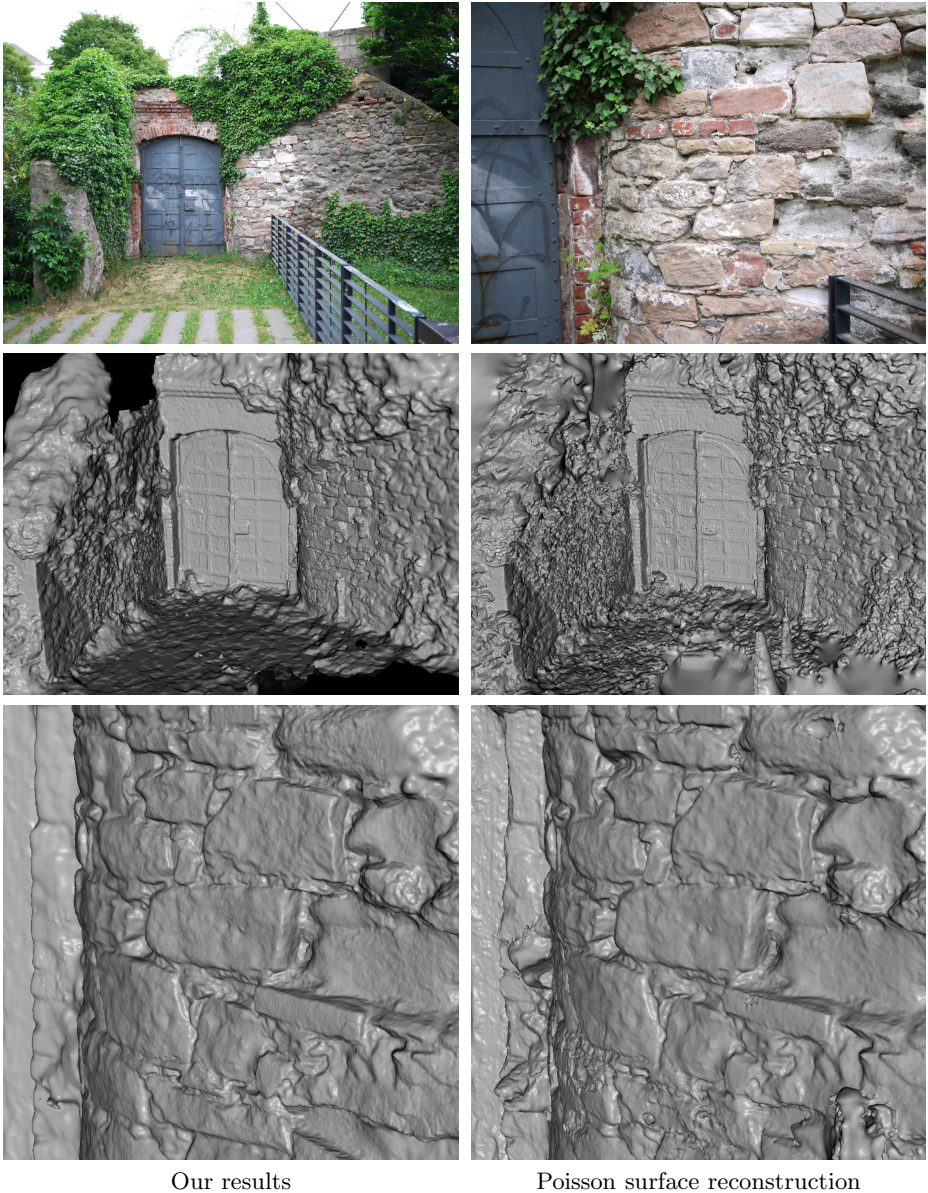
data set	sample points	vertices	octree level	comp. time	rel. variation in footprint
Temple	22 M	0.5 M	9	1 h	1.5
Kopernikus	32 M	3.3 M	10–12	1.5 h	38
Stone	43 M	4.3 M	8–14	4.5 h	75
Citywall	80 M	8.6 M	11–16	6 h	209



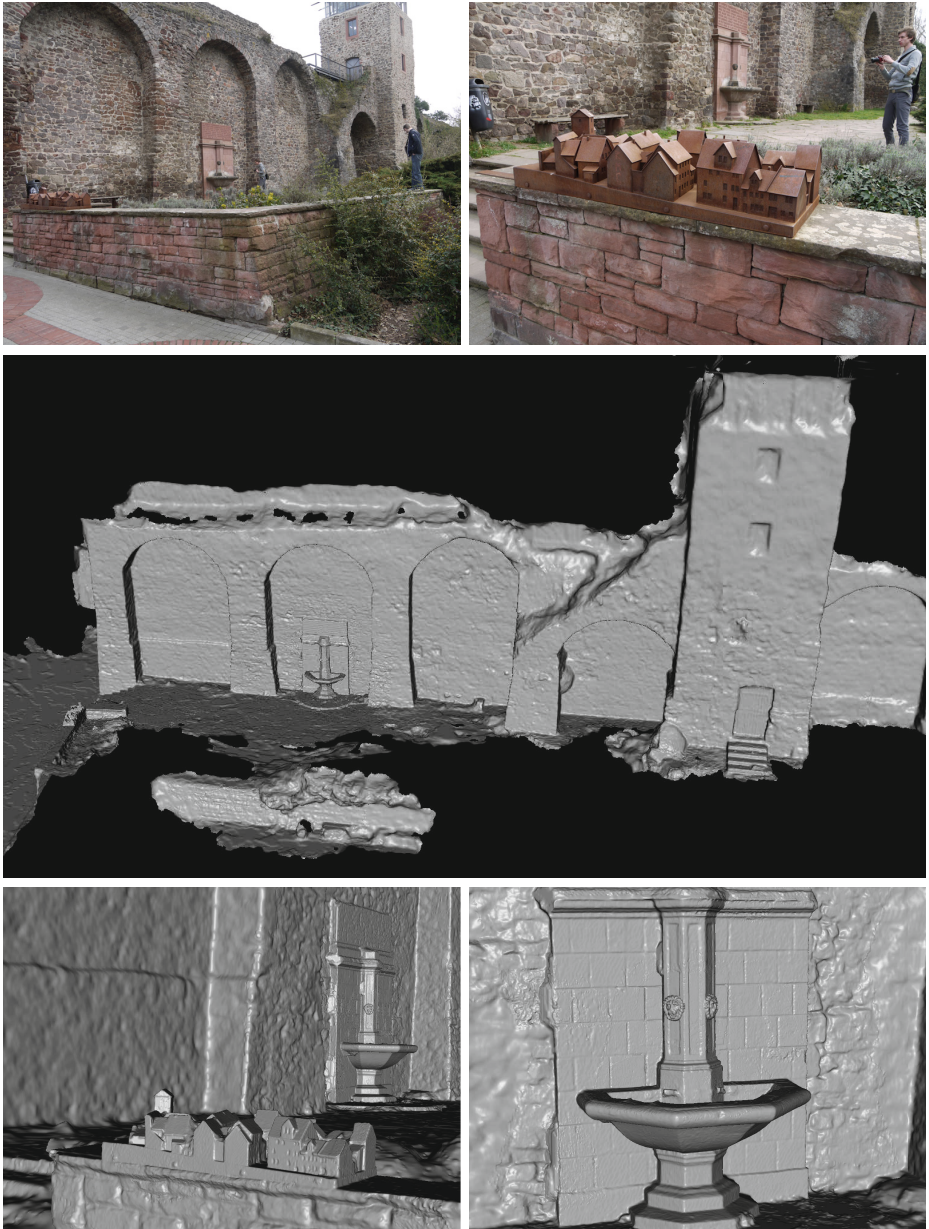
**Fig. 9.** An input image of the Temple data set (left) and a rendered view of our reconstructed model (right)

The Temple is a widely used standard data set provided by the Middlebury Multi-View Stereo Evaluation Project [20,17] and consists of 312 images showing a temple figurine. This data set can be considered to be single-resolution since all input images have the same resolution and distance to the object, resulting in the complete temple surface to be reconstructed on the same octree level in our algorithm. The reconstruction quality (Figure 9) is comparable to other state-of-the-art methods. We submitted reconstructed models created for a previous submission [18] for the TempleFull and the TempleRing variant (using only a subset of 47 images as input to the pipeline) to the evaluation. For TempleFull we achieved the best accuracy (0.36 mm, 99.7% completeness), for the TempleRing we achieved 0.46 mm at 99.1% completeness.

The stone data set consists of 117 views showing a region around a portal where one characteristic stone in the wall is photographed from a close distance leading to high-resolution sample points in this region. Overall we have a factor of 75 of variation in footprint sizes. In Figure 10 we compare our reconstruction with Poisson surface reconstruction [13]. In the overall view our reconstruction looks significantly better, especially on the ground where our method results in less noise. In the close-up view also Poisson surface reconstruction shows the fine details. Due to the fact that the sampling density is much higher around the particular stone Poisson surface reconstruction used smaller triangles for the reconstruction.



**Fig. 10.** *Top:* Example input images of the stone data set. *Middle + Bottom:* Comparison of our reconstruction (left) with Poisson surface reconstruction [13] (right). Although Poisson surface reconstruction does not take footprints into account the reconstruction shows fine details due to the higher sampling density. However, our surface shows significantly less noise and clutter.



**Fig. 11.** Top: Two input images of the Citywall data set. Middle: Entire model (color indicates the octree level, red is highest). Bottom: Close-ups of the two detailed regions.

The Citywall data set consists of 487 images showing a large area around a city wall. The wall is sampled with medium resolution, two regions though are sampled with very high resolution: the fountain in the middle and a small

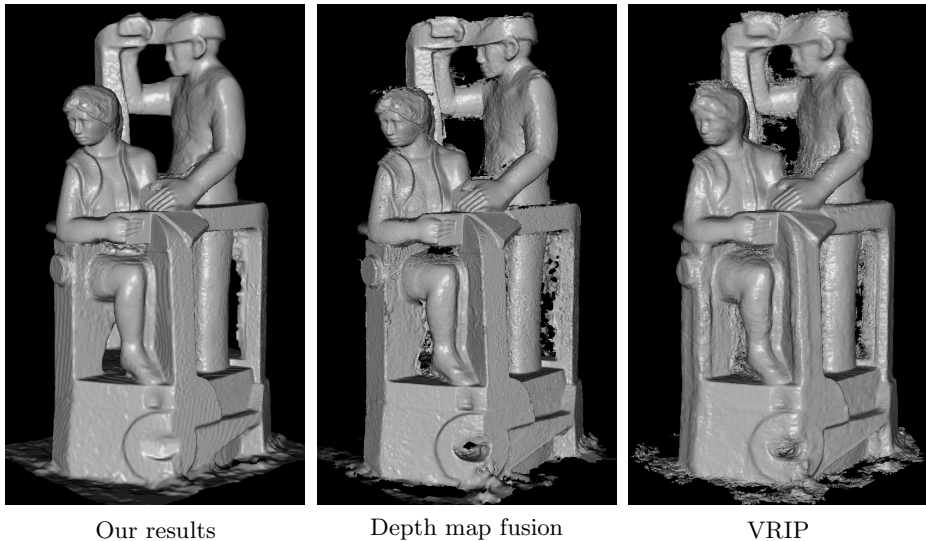


sculpture of a city to the left (Figure 11 top). Our multi-resolution method is able to reconstruct even fine details in the large scene where sample footprints differ up to a factor of 209. In consequence, the reconstruction spans six octree levels and detailed regions are triangulated about 32 times finer than low-resolution regions. The middle image of Figure 11 shows the entire mesh whereas the bottom images show close-ups of the highly detailed surface regions. One can even recognize some windows of the small buildings in the reconstructed geometry.

The Kopernikus data set (Figure 12) consists of 334 images showing a statue with a man and a women. The underlying surface geometry is particularly challenging due to its high genus. The data set is also multi-resolution in the sense that we took close-up views of the area around the hands. We compare our reconstruction against VRIP [5] and the depth map fusion by Fuhrmann and Gesele



**Fig. 12.** Two input images of the Kopernikus data set, the complete reconstructed model from two perspectives and a close-up of the wireframe showing the adaptively triangulated mesh



**Fig. 13.** Comparison of our reconstruction (*left*) with depth map fusion (*middle*) [6] and VRIP (*right*) [5]

[6] (Figure 1.3). It is clearly visible that our model contains significantly less noise and shows no clutter around the real surface. Also, the complex topology of the object is captured very well in comparison to the other methods. However, in regions with low-resolution geometry staircase artifacts are visible due to the surface extraction from a binary signed distance field. This is also visible in the wireframe rendering in Figure 1.2 (bottom right) showing the dense triangulation of the women’s face versus the coarse triangulation of the men’s upper body.

## 9 Conclusion and Future Work

We presented a robust surface reconstruction algorithm that works on general input data. To our knowledge, except for the concurrent work of Fuhrmann and Goesele [6], we are the first to take the footprint of a sample point into account during reconstruction. Together with a robust crust computation and an adaptive multi-resolution reconstruction approach we are able to reconstruct fine detail in large-scale scenes. We presented results comparable to state-of-the-art techniques on a benchmark data set and proved our superiority on challenging large-scale outdoor data sets and objects with complex topology. The triangle meshes are manifold and watertight and show an adaptive triangulation with smaller triangles in regions where higher details were captured.

In future work, we plan to explore other ways to distribute a sample point’s confidence over the volume, e.g., taking the direction to the sensor into account. This would allow us to better model the generally anisotropic error present in reconstructed depth maps.

**Acknowledgements.** This work was supported in part by the DFG Emmy Noether fellowship GO 1752/3-1.

## References

1. Alliez, P., Cohen-Steiner, D., Tong, Y., Desbrun, M.: Voronoi-based variational reconstruction of unoriented point sets. In: Proc. of Eurographics Symposium on Geometry Processing (2007)
2. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. of IEEE International Conference on Computer Vision (2003)
3. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004)
4. Boykov, Y., Veksler, O.: Graph cuts in vision and graphics: Theories and applications. In: *Handbook of Mathematical Models in Computer Vision* (2006)
5. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proc. of ACM SIGGRAPH (1996)
6. Fuhrmann, S., Goesele, M.: Fusion of depth maps with multiple scales. In: Proc. of ACM SIGGRAPH Asia (2011)
7. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2010)
8. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: Proc. of IEEE International Conference on Computer Vision (2007)
9. Habbecke, M., Kobbelt, L.: A surface-growing approach to multi-view stereo reconstruction. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2007)
10. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proc. of ACM SIGGRAPH (1992)
11. Hornung, A., Kobbelt, L.: Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2006)
12. Hornung, A., Kobbelt, L.: Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In: Proc. of Eurographics Symposium on Geometry Processing (2006)
13. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proc. of Eurographics Symposium on Geometry Processing (2006)
14. Labatut, P., Pons, J.P., Keriven, R.: Robust and efficient surface reconstruction from range data. *Computer Graphics Forum* (2009)
15. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: Proc. of ACM SIGGRAPH (1987)
16. Manson, J., Schaefer, S.: Isosurfaces over simplicial partitions of multiresolution grids. In: Proc. of Eurographics (2010)
17. Middlebury multi-view stereo evaluation, <http://vision.middlebury.edu/mview/>
18. Mücke, P., Klowsky, R., Goesele, M.: Surface reconstruction from multi-resolution sample points. In: Proc. of Vision, Modeling and Visualization (2011)
19. Project page, <http://www.gris.tu-darmstadt.de/projects/multires-surface-recon/>

20. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2006)
21. Shalom, S., Shamir, A., Zhang, H., Cohen-Or, D.: Cone carving for surface reconstruction. In: Proc. of ACM SIGGRAPH Asia (2010)
22. Sinha, S.N., Mordohai, P., Pollefeys, M.: Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: Proc. of IEEE International Conference on Computer Vision (2007)
23. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal sets for efficient structure from motion. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2008)
24. Vu, H.H., Keriven, R., Lbatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2009)
25. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust TV-L1 range image integration. In: Proc. of IEEE International Conference on Computer Vision (2007)

# Traffic Observation and Situation Assessment

Ralf Reulke<sup>1</sup>, Dominik Rueß<sup>2</sup>, Kristian Manthey<sup>2</sup>, and Andreas Luber<sup>3</sup>

<sup>1</sup> Humboldt-Universität zu Berlin, Computer Vision  
Unter den Linden 6, 10099 Berlin

<sup>2</sup> DLR German Aerospace Center, Institute of Robotics and Mechatronics

<sup>3</sup> DLR German Aerospace Center, Institute of Transportation Systems,  
Rutherfordstr. 2, 12489 Berlin, Germany  
reulke@informatik.hu-berlin.de,

{dominik.ruess,kristian.manthey,andreas.luber}@dlr.de

**Abstract.** Utilization of camera systems for surveillance tasks (e. g. traffic monitoring) has become a standard procedure and has been in use for over 20 years. However, most of the cameras are operated locally and data analyzed manually. Locally means here a limited field of view and that the image sequences are processed independently from other cameras. For the enlargement of the observation area and to avoid occlusions and non-accessible areas multiple camera systems with overlapping and non-overlapping cameras are used. The joint processing of image sequences of a multi-camera system is a scientific and technical challenge. The processing is divided traditionally into camera calibration, object detection, tracking and interpretation. The fusion of information from different cameras is carried out in the world coordinate system. To reduce the network load, a distributed processing concept can be implemented.

Object detection and tracking are fundamental image processing tasks for scene evaluation. Situation assessments are based mainly on characteristic local movement patterns (e.g. directions and speed), from which trajectories are derived. It is possible to recognize atypical movement patterns of each detected object by comparing local properties of the trajectories. Interaction of different objects can also be predicted with an additional classification algorithm.

This presentation discusses trajectory based recognition algorithms for atypical event detection in multi object scenes to obtain area based types of information (e.g. maps of speed patterns, trajectory curvatures or erratic movements) and shows that two-dimensional areal data analysis of moving objects with multiple cameras offers new possibilities for situational analysis.

**Keywords:** Traffic observation, multi-camera system, cooperative distributed vision, multi-camera orientation, multi-target tracking, situation assessment.

## 1 Introduction

Monitoring traffic at roads and road intersections is a well-known application for surveillance cameras. Video Image Detection Systems (VIDS) can derive traffic

parameters by means of image processing and pattern recognition methods. The temporal behavior of each object in the observation area can be described by trajectories.

For local applications (e. g. traffic light control) all vehicles, cyclists and pedestrians must be recognized. However, a special challenge is the detection of the interaction of road users. Zebra crossings on crossroads are a typical example where pedestrians and cyclists can collide with turning vehicles. The detection of such events can only be done with active or passive optical recognition methods.

The temporal behavior of every object in the observation area can be described by trajectories. The interaction between objects can also be determined from the trajectories.

Different views of the same area by more than one camera are necessary, due to limitations of single camera systems, resulting in line of sight blockage (occlusion) by other objects (e. g. cars, trees and traffic signs). Furthermore, a distributed cooperative multi-camera system (DCMCS) enables a significant enlargement of the observation area and a recording of activity and movement patterns based on trajectories.

The fusion of the derived data from different camera views is done on object or trajectory levels by a multi-target tracking approach. For this, only object specific features (center coordinate, size, color) are considered. To perform the fusion, these object features (e. g. the center coordinate) will be transformed from the camera coordinate system into a common world coordinate system. This information is then used as an additional measurement for the determination of the object trajectories. Prerequisite is an accurate common master clock for successful time tagging, and merging or tracking and an exact camera orientation.

The 2D view under certain observation conditions causes non-negligible shifts in the center coordinates of the same object. This is a reason for notable tracking errors. The use of wide baseline stereo methods can improve object detection, object characterization and tracking accuracy. A first approach for situation assessment is the assumption that normal traffic conditions can be derived from the analysis and clustering of the object trajectories in the observed region. The deviation from the typical clustered trajectory is detected as an abnormal event. The drawback is, that the complete trajectories for further evaluation frequently cannot be determined.

An alternative is a map based approach, which was developed in our group (see [38]). Semantic interpretation from trajectories or short-term tracklets can be derived by statistical evaluation. For that purpose, simple parameters derived from tracklets (for example speed and direction), are locally accumulated and statistically evaluated. Stronger deviations from the local statistics can be interpreted as atypical events and used for detection of atypical situations. The disadvantage of this method is that no interactions can be calculated between the object trajectories.

This contribution describes an expansion of the map based approach. For determination of object interactions an additional protocol was implemented,

which determines possible collisions between objects. The algorithm can be parallelized because of the independent local analysis.

This paper is organized as follows: First, review of existing systems for situation analysis and atypical event detection is given. Next is an overview of the DCMCS processing. In the fourth chapter the existing implementations are explained. Then former development in trajectory processing und situation assessment are summarized and recent evaluation and results are presented. This article closes with conclusion and outlook.

## 2 Situation Analysis and Atypical Event Detection Overview

Situation analysis in a road environment aims at the integration and interpretation of data from single or multiple sensor systems. The result of the process is a semantic description of the situation, applicable to higher level decision processes. Other areas for situation analysis may include surveillance applications, sport video analysis [22] or even customer tracking for marketing analysis [25]. In the following a short overview will be given.

An essential approach consists in the analysis of the trajectory. Jiang et. al. proposed in [21] an algorithm for clustering trajectories based on similarity measurements using the Bayesian information criterion (BIC) for model selection. It evaluates the trade-offs between the likelihood (quality of the model) and number of parameters (complexity of the model). The authors choose a Hidden Markov Model (HMM) to describe each trajectory and calculate the BIC for each. The HMM results represent frequent trajectories in the data set and can be used for detecting abnormal events. The advantage of this approach is, that this method is not sensitive to the absolute similarity values and that the number of clusters is calculated automatically.

Yao et. al. presents in [46] a contextual model for abnormal event detection based on a graphical representation of the trajectories augmented with spatio-temporal relations from a training scene. This relations model shows dependencies between a moving object and a semantic region (source, sink and path), the moving object itself (speed and acceleration), paired moving objects (vehicle distances) and interactions of multiple objects (pedestrians and cars moving over a crosswalk). The model is used for detecting abnormalities in subgraphs utilizing the log-likelihood ratio test. The results show that it is very useful as prior knowledge for tracking and abnormal event analysis.

Aköz and Karsligil [3] uses continuous HMM for clustering and the Expectation-Maximization algorithm for learning the model parameters. The approach relies on learning normal traffic flow using trajectory clustering techniques, then analyzing accident events by observing partial vehicle trajectories and motion characteristics. Unfortunately the number of clusters must be known in advance. Differentiating normal and abnormal events is done by defining descriptors and executes semantic decisions about traffic events and accident characteristics.

Whenever the partial log-likelihood values of observations towards a cluster declines significantly, an abnormal trajectory is detected.

Chiu and Tsai [11] introduce a macro-observation scheme for abnormal event detection. The scheme is based on energy maps which are created from the movement strength of each pixel in the training video. The movement strengths are weighted exponentially to emphasize the movement pattern of the most recent frames. The energy maps of the training observations are clustered into groups using hierarchical clustering. Abnormal events are detected by thresholding the minimum distance of the observation maps to the trained cluster centroids. The approach can easily be implemented for abnormality detection in daily life. It is not highly responsive to events that last only a few seconds.

Piciarelli and Foresti [35] present an online algorithm for clustering trajectories and building a tree-like structure for modeling spatio-temporal dependencies. The clusters are represented by a list of points and local variances. The distance between a point of a trajectory and a cluster is given by the Euclidean distance to the closest point in the cluster within a temporal sliding window. Once a cluster is created, it is arranged with all the other clusters in a tree. The edges of the tree are continuously annotated with the transition probability between the clusters on each tracked observation. The probability of a trajectory is the product of all transition probabilities of the tree-clusters it belongs to. Improbable trajectories are marked as abnormal. The problem here is that complete trajectories must be recorded, which cannot generally be expected.

Sillito and Fisher [44] introduced a semi-supervised method for learning normal trajectories using a Gaussian Mixture model. The trajectories are modelled by seven control points from an approximated cubic B-spline and classified by comparing the Mahalanobis distance to Gaussian mixture model with a threshold. If a trajectory exceeds that threshold, a human operator decides on the trajectory abnormality. In the case of normal trajectories, the classifier is updated. The combination of automatic classification and human operator improves the detection quality, since not every unusual trajectory is detected as abnormal.

### 3 Overview DCMCS Processing Approach

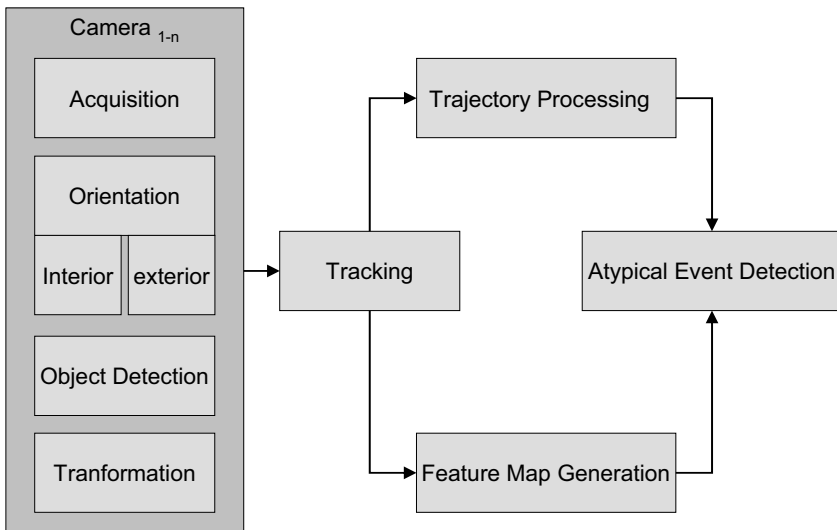
The approach proposed here is based on the use of a number of cameras (sensor network). The used cameras cover overlaid or adjacent observation areas. The same object can be observed using different cameras and camera views from different positions, time and observation angles. Using image processing methods the objects of interest can be detected in the images with background subtraction. A description of recent development in video processing and tracking can be found in the book from Maggio [29].

In order to enable tracking and fusion of the objects detected in the respective observation area, the image coordinates of these objects are converted to a common world coordinate system. In case of poor quality of the orientation parameters, the same objects, which were derived from different camera images,



will be observed at different places. To avoid false identification of these objects, high precision in coordinate transformation of the image into the object space is required. Therefore, an exact camera calibration (interior orientation) as well as knowledge of the position and view direction (exterior orientation) of the camera is necessary. If the camera positions are given in absolute geographical coordinates with known geographic direction, the detected objects can also be provided in real world coordinates.

The approach presented here can be separated into the following steps (Figure 1). In the brackets, the name of the boxes are given. We assume a number of  $n$  cameras. Firstly, after image acquisition for each camera objects have to be extracted from each frame of the video sequence (Acquisition, Object Detection). Next, the center coordinate of these objects have to be transformed onto a georeferenced world plane with known Z-coordinates (Orientation, Transformation). These processing steps are implemented very close to each camera. The results (center coordinates, object size, etc.) are transmitted to a distant processing unit. Here, the combined tracking occurs from the results of the distributed cameras. Afterwards the objects from all cameras are associated to trajectories (Tracking). This can be utilized to derive e.g. comprehensive traffic parameters (Trajectory Processing) and to characterize trajectories of individual objects (Atypical Event Detection) or for the generation of feature maps from tracklets (Feature Map Generation). These steps will be described in more detail below.



**Fig. 1.** Processing chain. The left block is camera related (from camera 1 to camera  $n$ ). The right part (beginning with tracking) is global with the data from all cameras.

### 3.1 Video Acquisition

In order to receive reliable and reproducible results, compact digital industrial cameras with standard interfaces and protocols (e.g. IEEE1394, GigE Vision Camera Interface) are used. All cameras have to be synchronized to a common master clock. While dealing with traffic surveillance and fast moving objects, the synchronization allows a maximum divergence of only a few milliseconds. In this approach the Network Time Protocol (NTP) is utilized to make all sensors use a common time axis. Continuous time synchronization with a maximum divergence of less than a millisecond between all cameras could be achieved using a local NTP-Server [40]. For acquisition of stereo data, the synchronization is implemented in hardware. A small trigger unit connects to the separate cameras. It can be set up to generate continuous triggering signals with different frequencies of 3, 7.5, 15 and 30 frames per second. The resulting maximum temporal difference of the corresponding images is here in the order of a few nanoseconds.

A centralized video server has been developed which receives all image data over TCP/IP. It is able to relay the images to any number of clients at any requested resolution and image quality, limited by the server computation power and server bandwidth only. There is no limitation for the number of clients with different requirements at the same time.

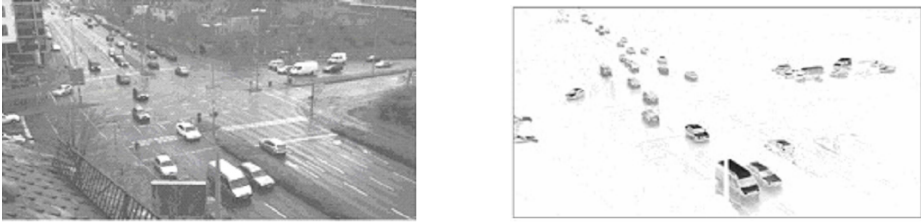
### 3.2 Background Estimation and Object Detection

To extract moving objects from an image sequence, different image processing libraries or programs (e.g. OpenCV or HALCON) can be utilized. The image processing and object detection algorithms used here are based on a background estimation in combination with other algorithms (e.g. shadow detection algorithm by Prati et al. [36]).

Different background algorithms are implemented and used in dependence of the applications. The background estimation provided by Javed et al. [20] is a Gaussian Mixture Models (GMM) and uses image gradients to differ between real objects in the foreground and regions that are falsely detected as foreground such as changes in illumination. Another background estimation was performed very similar to the approach of MacFarlane and Schofield [31]. The extraction is a combination of a Laplacian and a median average background. (For more details refer to [31]). A detected feature is considered as an object, if it has a minimum area of at least 30 pixels. For further processing the object is binarized. The resulting binarized object is frayed or disrupted. This effect is typically removed by a morphological closing.

A center point for every object can be determined by an ellipse fit. The ellipse encloses the object and remaining holes with a convex envelope. The resulting ellipses have shown to be quite stable in their shape, position and direction.

An example is shown in Figure 2.



**Fig. 2.** Acquired image (left), extracted objects (right)

### 3.3 Camera Calibration and Camera Orientation Determination

The determination of interior and exterior orientations are necessary preprocessing steps for further tasks. The interior orientation (camera calibration) is the determination of the camera model's parameters. Exact surveyed test fields were used to determine camera's interior and exterior orientations. Self-calibration, i.e. determination of interior orientation, is mostly used when test fields are not available during an on-the-job calibration. Since prior camera calibration is possible and the cameras broadly follows a perspective camera model, classic photogrammetric calibration can be used.

Cameras with very short focal lengths needs a replacement for the perspective model. The determination of exterior orientation is necessary for unique geometric relationships in overlapping regions. For the unambiguous determination of the geometric relationships between image and the world coordinate system, ground control points are necessary. The exterior orientation can be directly derived for each camera (see e.g. Reulke et al. [38]). If there are no control points, the relative orientation and bundle adjustment can be used. A particularly simple method is to use the projective transformation for a transition to a common world coordinate system. The mentioned methods will be described below.

**Camera Models.** The perspective projection with an additional distortion models is the most used camera model in literature and can model common types of cameras exactly. Despite that, there are many camera systems that cannot be calibrated at all or not precisely enough using this model. Especially the increasing usage of omnidirectional or wide-angle camera systems requires appropriate modelling.

Many different types of distortion models [8,9,7,27,24] extended by some additions proposed by [13] have been developed. Originally, these distortion models were implemented to compensate lens errors caused by physical effects or other manufacturing issues. The parameters (principal point, focal length and additional camera distortion in case of perspective camera model) can be determined using the test fields. The "Australis" software (based on a bundle block adjustment [16]) was used for parameter determination, which incorporates a 10 parameter model. With these parameters, the normalized (undistorted) image can be calculated from the distorted one.

If the error after the adjustment of the distortion model is too large, often the imaging model for the camera does not fit. Nevertheless, distortion models can be utilized to approximate non-perspective camera models by using them in addition to the classic perspective camera model. Even though this extended perspective projection yields sufficient accuracy at the expense of additional distortion parameters, it will probably fail when used for very wide fields of views. So called catadioptric or fish-eye lenses have a FOV beyond  $180^\circ$  and cannot be modeled using a perspective projection. However, there are parametric models that can approximate these types of lenses and other non-perspective lenses very well [43,47]. To avoid the decision for a proper camera model, a general camera model was introduced and evaluated by different authors for particular applications [5,12,17,33]. This generic model is suitable to calibrate the majority of commonly used camera systems including perspective and non-perspective types of projection. Despite its generic character, an adequate distortion model is often necessary to compensate additional lens errors.

The perspective camera model is characterized by the following radial distance function:

$$r = c \cdot \tan \theta \quad (1)$$

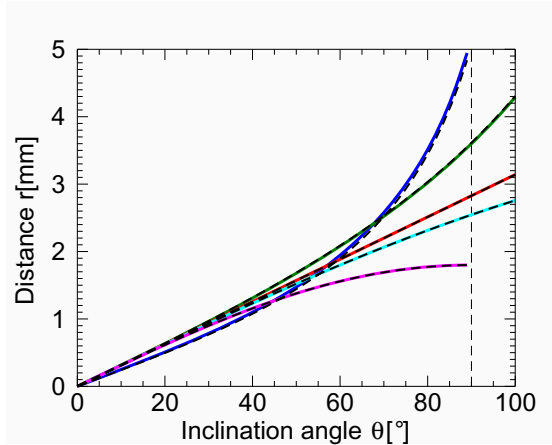
Where  $r$  is the radial distance of an image point starting from the principal point,  $c$  is the principal distance (focal length) and  $\theta$  is the inclination angle between the object ray and the optical axis. Other parametric models used, for e. g. fisheye camera systems, are:

$$\begin{aligned} \text{stereographic} &- r = 2c \cdot \tan \frac{\theta}{2} \\ \text{equidistant} &- r = c\theta \\ \text{equisolid-angle} &- r = 2c \cdot \sin \frac{\theta}{2} \\ \text{orthogonal} &- r = c \cdot \sin \theta \end{aligned} \quad (2)$$

To approximate a set of particular parametric models and to calibrate cameras which do not follow one of the mentioned parametric models a general model is needed. Here a polynomial with odd powers is appropriate to serve as a generic camera model since it approximates trigonometric functions [23,19]. The polynomial may be defined as follows:

$$r(\theta) = \sum_{i=1}^p k_i \theta^{2i-1} \quad (3)$$

By limiting the polynomial degree  $p$  to 3 or 4 the proposed generic camera model is able to replace many types of cameras with different parametric model. Furthermore, the field of views may exceed the problematic  $180^\circ$  degree angle limit of perspective and orthogonal projections. The parameter  $k_0$  corresponds to the classic focal length  $c$  (but may be change over the field of view).



**Fig. 3.** Camera Model Approximation: blue - perspective, green - stereographic, red - equidistant, cyan - equisolid angle, magenta - orthogonal, dashed black - accordant polynomial approximation

Figure 3 displays some arbitrary trigonometric camera models and their approximation by an polynomial model. Despite the perspective model, this generic approach is able to accurately approximate different trigonometric camera models. In this case a three parameter polynomial, i.e. 5th degree, was used.

**Exterior Orientation Determination.** For the determination of exterior orientation, i.e. orientation and position of the camera in world coordinate system, exact known ground control points (GCP) are necessary. Prominent image points must be assigned to known reference points. These reference points can be derived from known quantitative information like width of streets or markings on the lanes. Here, a high resolution ortho-image was used as reference map. Since no exact georeferencing is required, merely relative orientation or a much simpler projective transformation can be used.

The existing tracking concept (see section 3.6) is based on extracted objects, which are georeferenced to a world coordinate system. This concept allows the integration or fusion of additional data sources. Therefore, a transformation between image and world coordinates is necessary. Using collinearity equations (4), the world coordinates  $X, Y, Z$  can be derived from the image coordinates  $x', y'$ :

$$\begin{aligned}
 X &= X_0 + (Z - Z_0) \cdot \frac{r_{11} \cdot (x' - x_0) + r_{21} \cdot (y' - y_0) - r_{31} \cdot c}{r_{13} \cdot (x' - x_0) + r_{23} \cdot (y' - y_0) - r_{33} \cdot c} \\
 Y &= Y_0 + (Z - Z_0) \cdot \frac{r_{12} \cdot (x' - x_0) + r_{22} \cdot (y' - y_0) - r_{32} \cdot c}{r_{13} \cdot (x' - x_0) + r_{23} \cdot (y' - y_0) - r_{33} \cdot c}
 \end{aligned}
 \tag{4}$$

with the normalized image coordinates  $x', y'$ , the planar  $X, Y$  world coordinates (to be calculated) and the  $Z$ -component in world coordinates (to be known, can

be deduced by appointing a dedicated ground plane). The exterior orientation  $X_0, Y_0, Z_0$  (position of the perspective center in world coordinates) and  $r_{11}, r_{12}, \dots, r_{33}$  (elements of the rotation matrix), as well as the interior orientation  $x_0, y_0$  (image coordinates of the principal point) and  $c$  (focal length) have to be known.

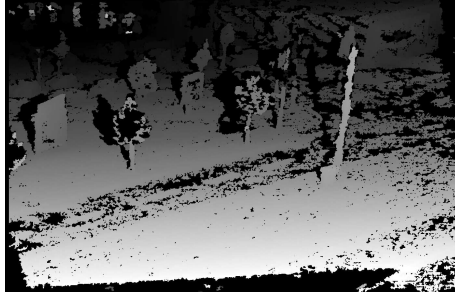
The calculation of the exterior orientation of a camera is based on previously measured ground control points (GCPs) e.g. with differential GPS. The accuracy of the points should be in the range of few centimeters. With these coordinates an approximate orientation can be deduced using DLT [21]. For improvement and elimination of erroneous GCPs the exterior orientation is calculated with the spatial resection algorithm. This algorithm was described previously [38]. This experimental setup was implemented at the intersection Rudower Chaussee / Wegedornstraße, Berlin (Germany). In another experimental setup consisting of two cameras, it has been installed at the intersection Rudower Chaussee / Brook-Taylor-Straße, Berlin (Germany). The observed area has an extent to about  $100 \times 100 \text{ m}^2$ . Figure 4 shows the original images taken from two different positions and in 5 the derived disparity map. The accuracy and the density of the derived disparity map is a good indicator for the sufficient accuracy of this approach. Before matching images a considerable good relative orientation has to be determined, which will be describe in the next section.



Fig. 4. Original images of the example scene

**Relative Orientation Determination.** If we consider cameras pairwise, then sometimes it is sufficient to determine the relative orientation (see e.g. Luhmann [27]) or fundamental matrix (see e.g. Ma [28]). Relative orientation determination can be done in different ways. We assume normalized image coordinates (after distortion correction), including uniform focal length.

If the exterior orientation has been determined with sufficient accuracy, the relative orientation between two cameras can be derived from this exterior orientation. For a more general discussion, let  $C_1$  and  $C_2$  be the absolute camera orientation in homogeneous coordinates for two cameras (both are of size  $4 \times 4$ ). Furthermore, let  $R$  be the unknown relative orientation:



**Fig. 5.** Disparity image, generated from images of two different observation positions

$$\begin{aligned} R \cdot C_1 &= C_2 \\ \Leftrightarrow R &= C_2 \cdot C_1^{-1}. \end{aligned} \quad (5)$$

If rotation and translation are assumed, only, the matrices can be written as:

$$\begin{pmatrix} R_r & \mathbf{t}_r \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_2 & \mathbf{t}_2 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} R_1^T & -R_1^T \mathbf{t}_1 \\ 0 & 1 \end{pmatrix}, \quad (6)$$

were  $R_i$  and  $\mathbf{t}_i$  are the respective rotation matrices and translation vectors. The advantage of this approach is the immediate integration into the world coordinate system and hence, the scale is already determined. (This procedure is equivalent to the determination of relative and absolute orientation.)

A second way of determining the relative orientation can be performed by a selection of (possibly manual) homologous point pairs. Using these, one can use the 8 point algorithm (see [28]) to determine a rough relative orientation. This is usually followed by a non-linear minimization approach, to obtain the exact relative orientation. This approach has been suggested in the *Computer Vision* domain, for more details or for less initial correspondences, for instance refer to [28]. This approach can be stabilized by use of RANSAC [14] which allows elimination of wrong or inaccurate correspondences.

The third way is contributed by the Photogrammetry domain. By utilizing the DLT, one obtains an initial linearly estimated relative orientation. *Bundle Adjustment* will refine this orientation, by minimizing the remaining root mean square error. For more details, refer for instance to [24]. The last two approaches are supposed to have slightly better accuracy as compared to the first approach. It is important to mention the different representation of rotation and translation in these two domains. However, it is rather simple to convert between those two. Let  $R$  be a camera's rotation matrix and  $t$  the respective projection center.



**Fig. 6.** Original (left), normalized (middle) and transformed image (right)

The Projection matrix, for use in the Computer Vision domain, is given by:

$$C = \begin{pmatrix} R & -Rt \\ 0 & 1 \end{pmatrix}. \quad (7)$$

All the described algorithms are implemented and used for different applications and frameworks. The accuracy of relative orientation determination can be determined e. g. from  $y$ -disparity (perpendicular to the  $x$ -disparity, which is correlated to the depth distance) for some characteristic points, seen in both images.

**Projective Transformation.** A particularly simple transformation from the image into the world coordinate system is the projective transformation. It is assumed that it is transformed from one plane to another plane. Especially in the street environment, this assumption is well fulfilled. The actual projective mapping from points  $\mathbf{x}$  of the camera image to points  $\mathbf{x}'$  of an ortho-image can be determined given a set  $D$  of manually selected corresponding homogeneous points. A projective mapping matrix  $P \in \mathbb{R}^{3 \times 3}$  which best fits the matches in  $L_2$ -sense can be determined by means of the SVD, for instance:  $P\mathbf{x} = \mathbf{x}'$ ,  $\forall(\mathbf{x}, \mathbf{x}') \in D$ . In the following figure 6 the described process is visualized.

### 3.4 Wide Baseline Stereo

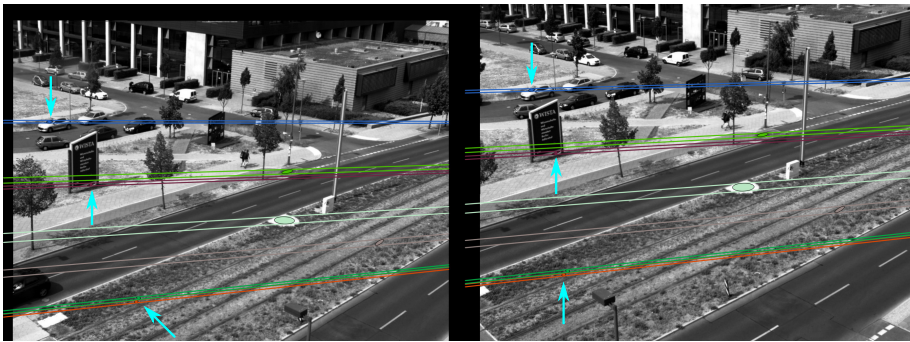
Due to perspective effects, fusion of 2D trajectories can lead to erroneous trajectories, due to displacement errors. Additional sources of errors are changes in illumination conditions, weather changes or in general outdoor setups. This may lead to many false positive objects in the background estimation process. Stereo reconstruction is independent of object movement and lighting conditions. But sometimes cameras are not or cannot be placed right next to each other.

Wide baseline stereo is a situation where the base length, that is the distance between the two camera centers, becomes significant with respect to the scene distance. In one of our scenarios, we have a camera pair with a base length of roughly 7 m. The distance to the scene starts with 15m and goes up to the horizon. To improve the object detection and tracking, additional constraints for wide baseline stereo were introduced.



Feature matching, that is detecting homologous point pairs, becomes more and more ambiguous the more the baseline increases. This is due to projective effects, different line of sight blockages (occlusions) in both cameras and possibly significant differences in illumination. To overcome this, additional constraints in epipolar geometry can be introduced, see [42]. Necessary are region features, similar to MSER features. They have already been proven to be good feature detectors for wide baseline situations, see [30].

The new constraints are based on a shape of the region features. MSER regions can be described with an ellipse, which is what was done in [42]. In projective space, ellipse tangents can be computed to any point. If the epipole is chosen as the point which both tangents have to cross, two new constraints arise: Both tangents map to tangents of the same feature in the second view, by use of the fundamental matrix  $F$ . Hence, both tangents or epipolar lines have to match, if we want correct correspondences. In figure 7, it some of these epipolar tangents can be seen and the fact that they match very well. In summary, the ellipse center, as well as the tangents have to match on the respective epipolar line. In [42] we have shown, that this reduces ambiguity of the matching process. These improvements have been utilized in the approach of Rueß et al. [41]. A future goal is to use these constraints to deduce a fast and more accurate relative orientation, as it can be shown, that only 3 correspondences are required when considering conics.



**Fig. 7. Outdoor Data Set.** Displayed are some of the correspondences after matching using the ellipse epipolar constraints. Smaller features are marked with an arrow. Additionally the tangency epipolar lines of the ellipses are shown.

### 3.5 Object Association in Consecutive Images

The following two sections deal with the object tracking. The simplest method (and the prerequisite for a complex tracking) is the object association.

To keep track of objects in between two consecutive images, we need to establish object correspondences. Objects from time  $t - 1$  have to be matched to

objects at time  $t$ . In general this can be done by generic appearance descriptors, which means looking for most similar objects compared to objects of the last time step. Again, object descriptions may include texture, shape but can also be position, same direction of movement etc.

This setup uses a template matching approach. The transformation to a world coordinate frame is reduced by projective mapping onto a plane (as described in 3.3). This transformation will distort objects in between frames but the temporal distance of two frames is very low. Thus, the objects still look very similar in consecutive time steps. This justifies the usage of the Normalized Cross Correlation (NCC).

### 3.6 Tracking, Trajectory Creation and Fusion

A number of objects are recognized for each image  $k$ . For the  $n$  objects a set of position data is available. The aim is to map the observation to an existing object and to update its state values describing this object, e.g. position or shape. There are different tracking algorithms. Tracking is done here by using a standard Kalman filter approach [46]. The basic idea consists of transferring supplementary information concerning the state into the filter approach in addition to the measurement. A forecast of the measuring results (prediction) is derived from earlier results of the filter. With that the approach is recursive. A map of the system state to the measurement vector has to be done in order to describe a complex state of an observed process:

$$\underline{Z}_k = \underline{H} \cdot \underline{X}_k + \beta_k + \varepsilon_k \quad (8)$$

with  $Z_k$  measurement of the sensor at time  $t_k$ ,  $X_k$  object state at  $t_k$ ,  $\beta_k$  unknown measurement offset,  $\varepsilon_k$  random measurement error,  $H$  Observation matrix and  $H \cdot X_k$  Measurement (object position).

Very important for the usability of the Kalman filter is the state model. A first approach for the state-vector is to assume for each object position, speed and acceleration only in X-axis and Y- direction ( $\underline{X}_k = [r_{xk} \ r_{yk} \ v_{xk} \ v_{yk} \ a_{xk} \ a_{yk}]'$ ). For the tasks described here, a more complex state model is reasonable. We used a state model consisting of position, speed (along the direction of moving), yaw angle and yaw rate ( $\underline{X}_k = [r_{xk} \ r_{yk} \ \psi_k \ v_{xyk} \ \omega_k]'$ ). This model is very useful when object are moving on curves, because this cannot be described only with acceleration as shown in the first approach. The observation matrix for the first model is given in equation 9 and in equation 10 for the second, respectively. The second filter is an *extended Kalman filter* (EKF), because the observation matrix depends on the current state. It has been shown, that using a *linear Kalman filter* for some first observations is useful. For 3D tracking a separate *Kalman filter* for the Z-axis was used. The measurement statistics will be described by uncorrelated white noise.

$$\underline{\underline{H}}_1 = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$\underline{\underline{H}}_2 = \begin{bmatrix} 1 & 0 & 0 & -v_{xyk} \sin(\psi_k) \Delta t & \cos(\psi_k) \Delta t & 0 & 0 \\ 0 & 1 & 0 & v_{xyk} \cos(\psi_k) \Delta t & \sin(\psi_k) \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The state transition model (plant model) is characterized by uniform motion. Since this one is idealized, an additional model error (predictions error, plant noise) was introduced.

$$\underline{X}_{k+1} = \underline{\underline{\Phi}}(\Delta t_k) \cdot \underline{X}_k + \underline{U}_k \tag{11}$$

with  $\underline{\underline{\Phi}}$  calculated from the movement model and  $\underline{U}_k$  plant noise.

If a (filtered) estimation is given at  $t_k$ , then the predicted state  $\underline{X}'_{k+1}$  at  $t_{k+1}$  is:

$$\underline{X}'_{k+1} = \underline{\underline{\Phi}}(\Delta t_k) \cdot \underline{\bar{X}}_k \quad t_{k+1} = t_k + \Delta t_k \tag{12}$$

The a posteriori state estimation is a linear combination of the a priori estimation and the weighted difference from the difference of forecast and measurement:

$$\underline{\bar{X}}_{k+1} = \underline{X}'_{k+1} + K(\underline{Z}_{k+1} - \underline{\underline{H}}\underline{X}'_{k+1}) \tag{13}$$

The initialization of the state-vector will be done from two consecutive images. The association of a measurement to an evaluated track is a statistical based decision-making process. It is implemented using the Kuhn-Munkres-Algorithm (Hungarian Method) which allows for solving bipartite graph assignment problems. Based on a feature distance this returns matched pairs of objects with an overall minimum sum of feature distances. Too large actual Euclidean space distances are rejected in advance.

Errors are related to clutter, object aggregation and splitting. The decision criteria minimize the rejection probability.

The coordinate projection mentioned in the last paragraph and the tracking process provides the possibility to fuse data acquired from different sensors. The algorithm is independent from the sensor as long as the data is referenced in a joint coordinate system and they share the same time frame.

The resulting trajectories are then used for different applications e. g. for the derivation of traffic parameters (TP).

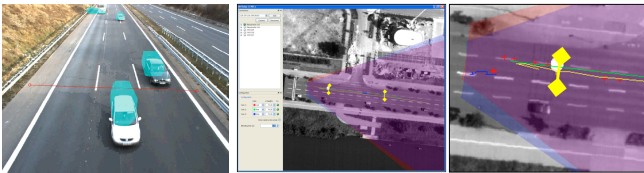
More details can be found in [41] and [45]. These are additional, slightly different implementations of the above described tracking system. In [41] a real 3D-tracking approach is realized and in [45], similar planar surveillance and 2D-merging techniques have been used.

## 4 Trajectory Analysis and Situation Assessment

The following examples were chosen to show the advantage of the trajectory based object description and situation assessment. Three different approaches will be described. Some results can be found in earlier publications ([37,39,38,40,34]). New are the investigation of interactions between traffic participants.

### 4.1 Counting Vehicles Using Virtual Detectors

The most common detection systems to measure traffic flow on public roads are inductive loops. Non-intrusive video-detection for traffic flow measurement is the primary alternative to conventional detectors. Often, the image processing is designed to analyze visual changes on a surface such as an induction loop. Tracking based approaches has some advantages. In particular object detection and accuracy of the derived data are much better. Image acquiring, processing and tracking was used as described in the pre-vius sections. After that, the different observations were merged to be able to track and summarize the results using, e. g., a curve fitting algorithm. Depending on the model, different types of data were generated for each vehicle (size, average speed, trajectory type) and saved to a database. In the paper [40], only when a vehicle passes the virtual induction loop was used and compared to that from the induction loop. Figure 8 shows a visualization of the results. The field of view (FOV) of the camera and the virtual loop are shown on an ortho-rectified image of the area. The derived object positions are marked as crosses and the trajectories as lines.



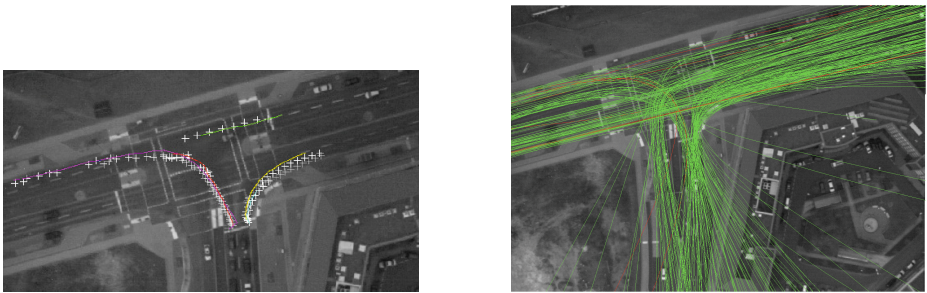
**Fig. 8.** Camera view with detected objects (left), FOV of the cameras and the virtual loop (middle), trajectory and an intersection with the virtual loop (right). Images are from [40].

## 4.2 Deterministic Analysis of Trajectories

A method for the deterministic description of trajectories was proposed by Pfeiffer [34]. For trajectories, the functional descriptions should be as simple as possible and permit a straightforward interpretation. Linear movements can be described by simple straights. But there are several possibilities of description for curve tracks by functional dependencies.

It exist a variety of suggestions of possible functions in the literature. Examples are Clothoid [26] or Splines [15], Cartesian polynomials fifth degree or Polarsplines [32].

Anderson [4] has proposed a description of tracks by hyperbolas. The advantage is that the derived parameters allows direct geometric interpretation and permit a classification and derivation of important features of the trajectories.



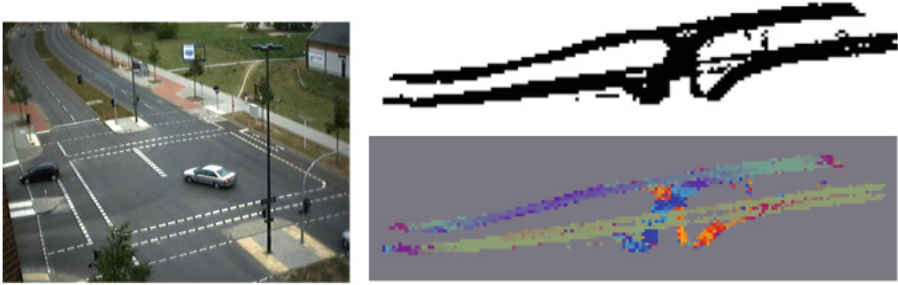
**Fig. 9.** Example tracks and trajectory fit, observed from a three cameras system (left) and a bunch of trajectories fit by hyperbolas (right)

Figure 9 shows an example of the implemented approach. The colored points and crosses are related to the trajectory, observed from different cameras. The hyperbola fit can be used for an automatic classification of right and left turns. In this case the angle  $\phi$  is positive or negative. With the calculated center  $(x_m, y_m)$  all four possibilities for right / left turning on the junction can be classified (see Pfeiffer [34]).

## 4.3 Trajectory Based Atypical Event Detection

For the detection of atypical events, trajectories have to be tagged, that fail to fit into the expected scheme, which was learned from a large number of previously recorded trajectories. For the deterministic approach, clustering of valid hyperbola parameters in a parameter space has been done and atypical trajectories could be sorted out by a threshold for the cluster distance. For the statistical analysis, a different approach was chosen.

It starts from the already described trajectory pieces (tracklets). These tracklets are conducted over a grid that was placed on a geocoded map. The grid

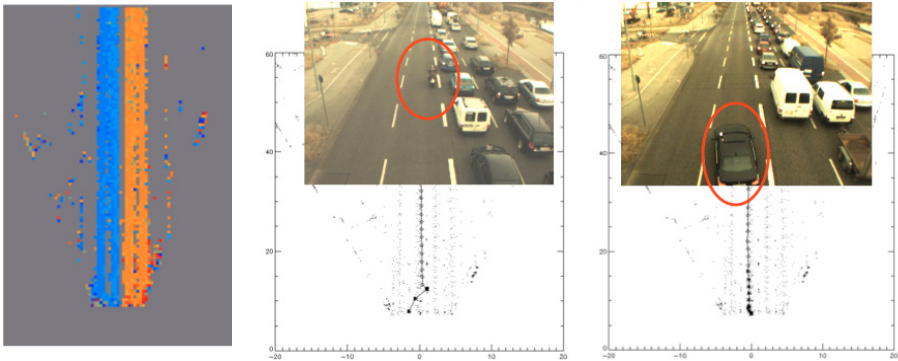


**Fig. 10.** Speed maps of an intersection (left). Detectors store activities in world coordinates in a  $1\text{ m} \times 1\text{ m}$  grid (upprt right). 2D histogram collects statistical values for the direction of the entire scene (lower right). The maps are generated from accepted trajectories and used for atypical trajectory detection.

size is  $1\text{ m} \times 1\text{ m}$  or less. For all tracklets that met boundary of a grid box, their statistical features were added into ortho-maps. At the example of speeds, figure 10 shows such a map for two different camera views right before fusion.

Atypical trajectories have been detected in our experiments by comparing currently tracked objects to the result of a map fusion and accumulate differences.

Figure 11 shows an example. Data about traffic direction where collected over a certain time (see figure left). Based on this direction map motorist driving against the traffic on motorways can be automatically detected.



**Fig. 11.** Direction map (left), detected wrong-way driver (middle and right)

A major drawback of the described method is that an interaction between road users cannot be detected without further assistance. In the following section, a method is described which provides these capabilities.

#### 4.4 Interaction and Dangerous Situation Analysis

The starting point is the observation of the scene in the world coordinate system. Each acquired image of the scene is transformed according to the section 3.3 (planar surveillance). As already described in section 4.3 the observation area is decompose into different sized areas. Unlike the previous sections 4.3, not only the statistics are determined but also performed the object recognition and tracking in these specified sub-areas. This makes a high degree of parallelization possible. In addition, another instance is introduced, which evaluates the local trajectories (tracklets) over the observation range.

All image patches  $p_i$  return objects  $o$ , including positions  $\mathbf{x}_{o,i}$  and velocities  $\mathbf{v}_{o,i}$ . Based on this information, a first possible danger analysis can be performed by computing the intersection of the two trajectory lines. If, for two objects  $o$  and  $o'$ , the solution  $\lambda$  and  $\lambda'$  of the linear system

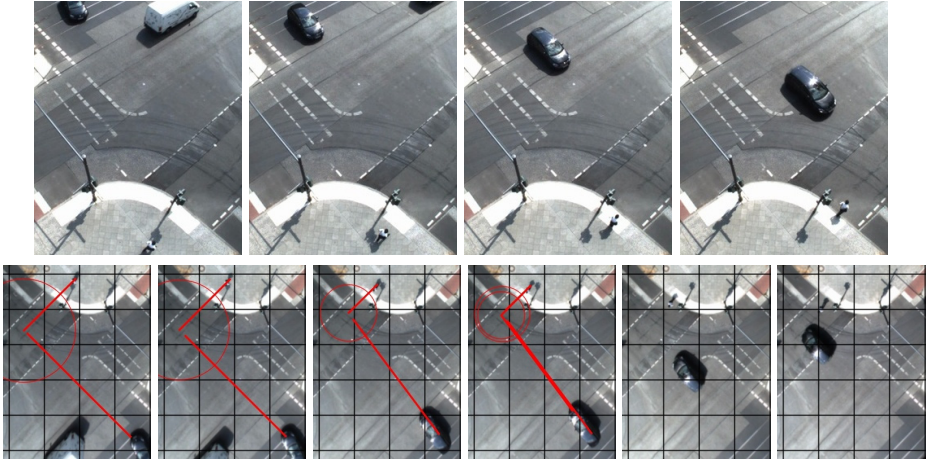
$$\mathbf{x}_{o,i} + \lambda \mathbf{v}_{o,i} = \mathbf{x}_{o',i'} + \lambda' \mathbf{v}_{o',i'}, \quad (14)$$

is real, there exists a section between the two paths. Here,  $\lambda$  and  $\lambda'$  represent the time (in number of frames) for the respective objects to reach that section. If, in addition, the difference of both times  $|\lambda - \lambda'|$  differs only a little (e. g. 10 frames, which equals 667 ms) then a collision is possible. A further improvement is to react only to possible collisions in nearby future, that is, one of the conditions  $\lambda < t_{\max}$  or  $\lambda' < t_{\max}$  has to be true.

#### 4.5 Improvements with SVM

In the planar surveillance system the goal was to find any kind of feature, which helps classifying the current situation. Each “smart camera” maps its input image to a plane, previously defined by the user (see section 3.3). Objects are extracted and matched between consecutive time steps (see section 3.5). This creates a movement vector for objects. Based on this movement, it is possible to detect situations like “near collision” or “area not allowed”. In general the objective was to classify as “hazardous/dangerous” or not. Due to several reasons, there are many false positive objects, as well as false positive “hazardous” situations.

To this end, we introduced supervised learning to our algorithms. Based on many different kinds of features, some of which are position, direction, speed, angle between objects, traffic lights, on/off street, we could show improvements. The “Support Vector Machine” (SVM, see [10], for instance) is able to model a set of training data by use of hyperplanes in feature space. It also has the ability to separate classes not only linearly and is known to produce very reasonable class boundaries. For this reason, we chose to employ the SVM for it can weigh different feature vectors on its own. An exemplary output can be seen in figure 12. It turns out, that the number of false positives can be reduced significantly whilst keeping the number of true positives high. More details can be found in [18].



**Fig. 12. Dangerous situation.** Upper row: original frames of a pedestrian vs car sequence. Lower row: possible output of danger analysis with possible collisions marked with a circle (radius: time in frames to possible collision).

## 5 Conclusion and Outlook

The presented approach for a multi-camera multi-object tracking and surveillance system has been implemented and tested. Thus, it could be shown that relevant surveillance tasks and automatic scene description can be automatically assisted based on video detection, tracking and trajectory analysis.

This is a necessary step for the future of next generation surveillance systems. However, detection errors and tracking problems can deteriorate the trajectory data. This leads to less usable trajectories for analysis or less reliable alarm rates for the operators. Methods to recognize object detection errors and deteriorated trajectories to stitch them together are key factors in the current and future work. Furthermore, the intelligent evaluation of feature maps for atypical trajectory detection will be expanded.

**Acknowledgements.** Dominik Rueß and Kristian Manthey would like to acknowledge financial support of the Helmholtz Research School on Security Technologies.

## References

1. Abdel-Aziz, Y.I.: Photogrammetric Potential of Non Metric Cameras. PhD thesis, University of Illinois, Photogrammetric potential of non metric cameras (1974)
2. Abdel-Aziz, Y.I., Karara, H.M.: Direct linear transformation into object space coordinates in close-range photogrammetry. In: Symposium on CloseRange Photogrammetry, Urbana, Illinois, pp. 1–18 (1971)



3. Aköz, Ö., Elif Karşlıgil, M.: Video-based traffic accident analysis at intersection using partial vehicle trajectories. In: Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR 2010, p. 335 (2010)
4. Anderson, B., Moor, J.: Optimal filtering. Prentice-Hall, Inc., Enlewood Cliffs (1979)
5. Basu, A., Licardie, S.: Alternative models for fish-eye lenses. *Pattern Recognition* 16(4), 433–441 (1995)
6. Blackman, S.S.: Multiple-target tracking with radar applications. Artech House, MA (1986)
7. Brown, D.C.: Close range camera calibration. *Photogrammetric Engineering* 37(8), 855–866 (1971)
8. Brown, D.C.: An advanced plate reduction for photogrammetric cameras. Technical report, Air Force Cambridge Research Laboratories (1964)
9. Brown, D.C.: Decentering distortion of lenses. *Photogrammetric Engineering* 32(7), 444–462 (1965)
10. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 121–167 (1998)
11. Chiu, W.-Y., Tsai, D.-M.: A Macro-Observation Scheme for Abnormal Event Detection in Daily-Life Video Sequences. *EURASIP Journal on Advances in Signal Processing* 2010, 1–20 (2010)
12. Claus, D., Fitzgibbon, A.W.: A rational function lens distortion model for general cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 213–219 (June 2005)
13. El-Hakim, S.F.: Real-time image meteorology with ccd cameras. *Photogrammetric Engineering and Remote Sensing* 52(11), 1757–1766 (1986)
14. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
15. Forbes, A.B.: Least-squares best fit geometric elements. Technical report, National Physical Laboratory of Great Britain, NPL-report, DITC 140/89 (1989)
16. Fraser, C.S., Edmundson, K.L.: Design and implementation of a computational processing system for off-line digital close-range photogrammetry. *ISPRS Journal of Photogrammetry and Remote Sensing* 55(2), 94–104 (2000), doi:10.1016/S0924-2716(00)00010-1
17. Gennery, D.B.: Generalized camera calibration including fish-eye lenses. *International Journal of Computer Vision* 68, 239–266 (2002)
18. Heideklang, R.: Employing a support vector machine to detect hazardous traffic situations, Student's thesis, Humboldt-Universität zu Berlin (2011)
19. Heikkilä, J.: A polynomial camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(8), 1335–1340 (2006)
20. Javed, O., Shafique, K., Shah, M.: A hierarchical approach to robust background subtraction using color and gradient information. In: IEEE Workshop on Motion and Video Computing, p. 22 (2002)
21. Jiang, F., Wu, Y., Katsaggelos, A.K.: Abnormal Event Detection from Surveillance Video by Dynamic Hierarchical Clustering. In: 2007 IEEE International Conference on Image Processing, pp. 1:V – 145–V – 148 (2007)
22. Jiang, S., Ye, Q., Gao, W., Huang, T.: A new method to segment playfield and its applications in match analysis in sports video. In: MULTIMEDIA 2004: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 292–295. ACM Press, New York (2004)

23. Kannala, J., Brandt, S.S.: A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions. Pattern Analysis and Machine Intelligence* 28, 1335–1340 (2006)
24. Kraus, K.: *Photogrammetrie 1*, 7th edn. Gruyter Verlag (2004) ISBN-10: 3110177080, ISBN-13: 978-3110177084
25. Leykin, A., Tuceryan, M.: A vision system for automated customer tracking for marketing analysis: Low level feature extraction. In: *International Workshop on Human Activity Recognition and Modeling*, pp. 1–7 (2005)
26. Liscano, R., Green, D.: Design and implementation of a trajectory generator for an indoor mobile robot. In: *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems*. Tsukuba, Japan, pp. 380–385 (1989)
27. Luhmann, T., Robson, S., Kyle, S., Harley, I.: *Close-Range Photogrammetry*. Whittles Publishing (2006)
28. Ma, Y., Soatto, S., Kořecká, J., Shankar Sastry, S.: *An Invitation to 3-D Vision*. Springer (2004)
29. Maggio, E., Cavallaro, A.: *Video Tracking: Theory and Practice*. John Wiley & Sons (2011)
30. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* 22(10), 761–767 (2004); *British Machine Vision Computing* 2002
31. McFarlane, N.J.B., Schofield, C.P.: Segmentation and tracking of piglets in images. *Machine Vision and Applications* 8(3), 187–193 (1995)
32. Nelson, W.L.: Continuous steering-function control of robot carts. *IEEE Transactions on Industrial Electronics* 36(3), 330–337 (1989)
33. Orekhov, V., Abidi, B., Broaddus, C., Abidi, M.: Universal camera calibration with automatic distortion model selection. In: *IEEE International Conference on Image Processing*, vol. 6, pp. 397–400 (2007)
34. Pfeiffer, D., Reulke, R.: Trajectory-based scene description and classification. In: Stilla, U., Rottensteiner, F., Paparoditis, N. (eds.) *Object Extraction for 3D City Models, Road Databases and Traffic Monitoring*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, vol. 38, pp. 41–46 (2009)
35. Piciarelli, C., Foresti, G.: On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters* 27(15), 1835–1842 (2006)
36. Prati, A., Mikic, I., Grana, C., Trivedi, M.M.: Shadow detection algorithms for traffic flow analysis: a comparative study. In: *Proc. IEEE Intelligent Transportation Systems Conf.*, pp. 340–345 (2001)
37. Reulke, R., Bauer, S., Döring, T., Meysel, F.: Traffic surveillance using multi-camera detection and multi-target tracking. In: Cree, M. (ed.) *Image and Vision Computing New Zealand 2007*. University of Waikato, New Zealand (December 2007)
38. Reulke, R., Meysel, F., Bauer, S.: Situation Analysis and Atypical Event Detection with Multiple Cameras and Multi-Object Tracking. In: Sommer, G., Klette, R. (eds.) *RobVis 2008*. LNCS, vol. 4931, pp. 234–247. Springer, Heidelberg (2008)
39. Reulke, R., Bauer, S., Spangenberg, R.: Multi-camera detection and multi-target tracking. To be published in *VISAPP - 3rd International Conference on Computer Vision Theory and Applications* (2008)
40. Reulke, R., Meffert, B., Piltz, B., Bauer, S., Hein, D., Hohloch, M., Kozempel, K.: Long-term investigations of quality and reliability of the video image detection system m3. In: *International Workshop on Traffic Data Collection and its Standardization* (2008)

41. Rueß, D., Manthey, K., Reulke, R.: An accurate 3d feature tracking system with wide-baseline stereo smart cameras. In: 2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), pp. 1–6 (August 2011)
42. Rueß, D., Reulke, R.: Ellipse Constraints for Improved Wide-Baseline Feature Matching and Reconstruction. In: Aggarwal, J.K., Barneva, R.P., Brimkov, V.E., Koroutchev, K.N., Korutcheva, E.R. (eds.) IWACA 2011. LNCS, vol. 6636, pp. 168–181. Springer, Heidelberg (2011)
43. Schneider, D., Schwalbe, E., Maas, H.-G.: Validation of geometric models for fisheye lenses. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(3), 259–266 (2009)
44. Sillito, R.R., Fisher, R.B.: Semi-supervised Learning for Anomalous Trajectory Detection. In: *BMVC*, vol. 27, pp. 1025–1044 (October 2008)
45. Treutner, N., Hellwig, S., Rueß, D.: A framework for people tracking and situation evaluation in multi-camera outdoor environments. In: Paul, L., Stanke, G., Pochanke, M. (eds.) *3D-Nordost, GFaI* (2011)
46. Yao, B., Wang, L., Zhu, S.-C.: Learning a Scene Contextual Model for Tracking and Abnormality Detection. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (June 2008)
47. Zhang, Z.: A flexible new technique for camera calibraion. Technical report, Microsoft Research, A Flexible New Technique For Camera Calibraion (1998)

# Author Index

- Abraham, Steffen 329
- Balcan, Doru C. 131
- Banitsas, Konstantinos 27
- Bao, Sid Yingze 376
- Bartelsen, Jan 285
- Brilakis, Ioannis 151
- Bronstein, Alexander M. 177
- Bronstein, Michael M. 177
- Brostow, Gabriel 329
- Dai, Fei 151
- Debard, Glen 356
- Dejaeger, Eddy 356
- Dellaert, Frank 131
- Deschodt, Mieke 356
- Dragon, Ralf 110
- Fenzi, Michele 110
- Förstner, Wolfgang 329
- Gall, Juergen 243, 305
- Gehrig, Stefan 27, 329
- Goedemé, Toon 356
- Goesele, Michael 398
- Hermann, Simon 52
- Hirschmüller, Heiko 285
- Imiya, Atsushi 78, 329
- Jähne, Bernd 329
- Jian, Yong-Dian 131
- Karsmakers, Peter 356
- Kimmel, Ron 177
- Klette, Reinhard 52
- Klose, Felix 329
- Klowsky, Ronny 398
- Kobbelt, Leif 191
- Koch, Reinhard 212
- Kondermann, Daniel 329
- Kuhn, Andreas 285
- Leal-Taixé, Laura 1, 305
- Leibe, Bastian 191
- Luber, Andreas 419
- Magnor, Marcus 329
- Manthey, Kristian 419
- Mayer, Helmut 285, 329
- Mester, Rudolf 329
- Milisen, Koen 356
- Mochizuki, Yoshihiko 78
- Morales, Sandino 52
- Mücke, Patrick 398
- Ostermann, Jörn 110
- Pajdla, Tomas 329
- Pfeiffer, David 27
- Pons-Moll, Gerard 1, 305
- Radopoulou, Stefania-Christina 151
- Raviv, Dan 177
- Razavi, Nima 243
- Reulke, Ralf 329, 419
- Rosenhahn, Bodo 1, 110, 264, 305
- Rueß, Dominik 419
- Sattler, Torsten 191
- Savarese, Silvio 376
- Schneider, Nicolai 27
- Sedlazeck, Anne 212
- Siberski, Wolf 110
- Sochen, Nir 177
- Sörgel, Uwe 264
- Tuytelaars, Tinne 356
- Van Gool, Luc 243
- Vanrumste, Bart 356
- Vlaeyen, Ellen 356
- Wegner, Jan D. 264
- Zimmer, Henning 329