DGK Veröffentlichungen der DGK
Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 792

Mathias Rothermel

Development
# of a SGM-Based Multi-View Reconstruction Framework for Aerial Imagery

München 2016

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5204-8

DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 792

Development
# of a SGM-Based Multi-View Reconstruction Framework for Aerial Imagery

Von der Fakultät Luft- und Raumfahrttechnik und Geodäsie
der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von

Dipl.-Ing. Mathias Rothermel

aus Stuttgart

München 2016

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5204-8

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)
Alfons-Goppel-Straße 11 ● D – 80 539 München
Telefon +49 – 89 – 23 031 1113 ● Telefax +49 – 89 – 23 031 - 1283 / - 1100
e-mail post@dgk.badw.de ● http://www.dgk.badw.de

Hauptberichter: Prof. Dr.-Ing. habil. Dieter Fritsch
Mitberichter: Prof. Dr Luc Van Gool.

Tag der mündlichen Prüfung: 11.11.2016

Diese Dissertation ist auf dem Server der DGK unter <http://dgk.badw.de/>
sowie auf dem Server der Universität Stuttgart unter <http://elib.uni-stuttgart.de/opus/doku/e-diss.php>
elektronisch publiziert

© 2016 Bayerische Akademie der Wissenschaften, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen.

ISSN 0065-5325 ISBN 978-3-7696-5204-8

Abstract

Advances in the technology of digital airborne camera systems allow for the observation of surfaces with sampling rates in the range of a few centimeters. In combination with novel matching approaches, which estimate depth information for virtually every pixel, surface reconstructions of impressive density and precision can be generated. Therefore, image based surface generation meanwhile is a serious alternative to LiDAR based data collection for many applications. Surface models serve as primary base for geographic products as for example map creation, production of true-ortho photos or visualization purposes within the framework of virtual globes. The goal of the presented theses is the development of a framework for the fully automatic generation of 3D surface models based on aerial images - both standard nadir as well as oblique views. This comprises several challenges. On the one hand dimensions of aerial imagery is considerable and the extend of the areas to be reconstructed can encompass whole countries. Beside scalability of methods this also requires decent processing times and efficient handling of the given hardware resources. Moreover, beside high precision requirements, a high degree of automation has to be guaranteed to limit manual interaction as much as possible. Due to the advantages of scalability, a stereo method is utilized in the presented thesis. The approach for dense stereo is based on an adapted version of the semi global matching (SGM) algorithm. Following a hierarchical approach corresponding image regions and meaningful disparity search ranges are identified. It will be verified that, dependent on undulations of the scene, time and memory demands can be reduced significantly, by up to 90% within some of the conducted tests. This enables the processing of aerial datasets on standard desktop machines in reasonable times even for large fields of depth. Stereo approaches generate disparity or depth maps, in which redundant depth information is available. To exploit this redundancy, a method for the refinement of stereo correspondences is proposed. Thereby redundant observations across stereo models are identified, checked for geometric consistency and their reprojection error is minimized. This way outliers are removed and precision of depth estimates is improved. In order to generate consistent surfaces, two algorithms for depth map fusion were developed. The first fusion strategy aims for the generation of 2.5D height models, also known as digital surface models (DSM). The proposed method improves existing methods regarding quality in areas of depth discontinuities, for example at roof edges. Utilizing benchmarks designed for the evaluation of image based DSM generation we show that the developed approaches favorably compare to state-of-the-art algorithms and that height precisions of few GSDs can be achieved. Furthermore, methods for the derivation of meshes based on DSM data are discussed. The fusion of depth maps for 3D scenes, as e.g. frequently required during evaluation of high resolution oblique aerial images in complex urban environments, demands for a different approach since scenes can in general not be represented as height fields. Moreover, depths across depth maps possess varying precision and sampling rates due to variances in image scale, errors in orientation and other effects. Within this thesis a median-based fusion methodology is proposed. By using geometry-adaptive triangulation of depth maps depth-wise normals are extracted and, along the point coordinates are filtered and fused using tree structures. The output of this method are oriented points which then can be used to generate meshes. Precision and density of the method will be evaluated using established multi-view benchmarks. Beside the capability to process close range datasets, results for large oblique airborne data sets will be presented. The report closes with a summary, discussion of limitations and perspectives regarding improvements and enhancements. The implemented algorithms are core elements of the commercial software package SURE, which is freely available for scientific purposes.

Kurzfassung

Moderne digitale Luftbildkamerasysteme ermöglichen die Beobachtung von Oberflächen mit Abtastraten im Bereich weniger Zentimeter. In Kombination mit neuen Verfahren der Bildzuordnung, welche Tiefeninformation für nahezu jedes Pixel schätzen, können somit Oberflächenrekonstruktionen mit beeindruckender Genauigkeit und Dichte generiert werden. Oberflächenmodelle dienen als primäre Grundlage für geographisch Produkte wie beispielsweise zur Erstellung von Karten, Orthophotos oder zu Visualisierungszwecken im Rahmen virtueller Globen. Ziel der vorliegenden Arbeit ist die Entwicklung eines Verfahrens für die vollautomatische Generierung von 3D Obeflächen- modellen basierend auf Luftbildern - sowohl für Nadir- als auch Schrägbildkonfigurationen. Dieses Problem beinhaltet einige Herausforderungen. Zum einen ist die Größe von Luftbildern beachtlich und die Ausdehnung rekonstruierter Gebiete kann komplette Länder umfassen. Dies verlangt neben der Skalierbarkeit der Verfahren auch Schnelligkeit und Effizienz im Umgang mit den gegebenen Hardwareresourcen. Des weiteren müssen neben hohen Präzissionsansprüchen, die eingesetzten Verfahren einen hohen Automatisierungsgrad aufweisen, um manuelle Interaktion weitestgehend zu vermeiden. Aufgrund der Vorteile bezüglich Skalierbarkeit kommt in der vorliegenden Arbeit ein Stereoverfahren zum Einsatz. Die vorgestellte Methode zur dichten Stereorekonstruktion basiert auf einer Erweiterung des Semi-Global-Matching Algorithmus. Einem hierarchischen Ansatz folgend werden dabei sukzessive sowohl korrespondierende Bildausschnitte als auch sinnvolle Disparitätssuchräume ermittelt. In Untersuchungen wird aufgezeigt, dass so, je nach Tiefenvarianzen der Szene, Speicher- und Zeitaufwand um bis zu 90% reduziert werden koennen. Stereo-Verfahren generieren typischerweise Disparitäts- oder Tiefenkarten, in welchen Tiefeninformation redundant vorliegt. Um diese Redundanz zu nutzen, wird in der vorliegenden Arbeit eine Methode zur Verfeinerung der Stereokorrespondenzen vorgestellt. Dabei werden redundante Beobachtungen zwischen Stereomodellen identifiziert, auf geometrische Konsistenz geprüft und anschließend deren Reprojektionsfehler minimiert. So können zum einen Ausreißer eliminiert und zum anderen die Genauigkeit einzelner Tiefenkarten verbessert werden. Um konsistente Oberflächen zu generieren, wurden desweiteren zwei Algorithmen zur Fusion von Tiefenkarten entwickelt. Das erste Fusionsverfahren dient der Generierung von digitalen Oberflächenmodelles(DOM). Das vorgestellte Verfahren verbessert bisherige Methoden hinsichtlich Robustheit an Tiefendiskontinuitäten, beispielsweise in Bereichen von Dachkanten. Anhand eines Benchmarks für die DOM-Generierung wird aufgezeigt, dass das entwickelte Verfahren hinsichtlich Genauigkeit und Prozessierungszeit mit dem Stand der Technik konkurrieren und Höhengenauigkeiten im Bereich weniger GSDs erzielt werden können. Des weiteren werden Methoden zur Ableitung von Vermaschungen von DOM-Daten diskutiert. Die Fusion von Tiefenkarten für 3D Szenen erfordert eine andere Herangehensweise, da die Szene nicht als Höhenfeld dargestellt werden kann. Vielfach weisen Tiefen aufgrund von Varianzen des Bildmaßstabs, Orientierungfehlern und anderer Effekte häufig unterschiedliche Genauigkeiten auf. In dieser Dissertation wird ein Verfahren der median-basierten Fusionierung vorgestellt. Dabei werden unter Verwendung geometrieadaptiver Vermaschungen Normalen in Tiefenkarten extrahiert und mittels Baumstrukturen fusioniert und gefiltert. Das Verfahren generiert orientierte Punkte, welche anschließend vermascht werden können. Ergebnisse werden hinsichtlich Genauigkeit und Dichte mittels der gängingen Mehrbildstereo-Benchmarks verifiziert. Die vorliegende Arbeit schließt mit einer Zusammenfassung, Beschreibung von Limitierungen der entwickelten Verfahren und einem Ausblick. Die implementierten Algorithmen sind Kernelemente der kommerziellen Softwarelösung SURE, welche für wissenschaftliche Nutzung frei verfügbar ist.

Contents

1 Introduction **9**
1.1 Motivation 9
1.2 Objectives 10
1.3 Main Contributions 11
1.4 Outline 12

2 Related Work **13**
2.1 Multi-View Systems 13
  2.1.1 Reconstruction Algorithms 13
  2.1.2 Scene Representations 14
  2.1.3 Photo Consistency Measures 14
  2.1.4 Visibility Models 16
  2.1.5 Shape Priors and Optimization Concepts 16
  2.1.6 MVS with Regard to the Generation of Elevation Data 19
2.2 Dense Stereo 20
  2.2.1 Problem Formulation 20
  2.2.2 Disparity Refinement 21
  2.2.3 Filter Techniques in Dense Stereo 21
2.3 Consistent Surface Models From Point Clouds and Depth Maps 22

3 Overview of the Reconstruction Process **25**
3.1 Model Selection 26
3.2 Depth Map Computation 27
3.3 Depth Map Fusion 27

4 Generation of Depth Maps **29**
4.1 Rectification of Calibrated Image Pairs 30
  4.1.1 Examples 30
4.2 SGM-based Dense Matching 30
4.3 Multi Baseline Triangulation 41
  4.3.1 Correspondence Linking 41
  4.3.2 Geometric Consistency Filters 42
  4.3.3 Multi-Baseline Forward Intersecion 44
  4.3.4 Example for Multi-Baseline Stereo 46
4.4 Results 46
  4.4.1 Comparision of classical SGM and tSGM 46
  4.4.2 Evaluation of Multi-Baseline Stereo 50

5 Disparity Map Fusion for 2.5D Model Generation **57**
5.1 Fusion Strategy for 2.5D Elevation Models 57
5.2 Interpolation of Elevation Maps 58

5.3 Results 59
5.3.1 Processing Times 60
5.3.2 Differences and Precision of Benchmark DSMs 62
5.4 Meshing of 2.5D Elevation Maps 64
5.4.1 Meshing of Elevation Data Using Restricted Quad Trees 66
5.4.2 Re-meshing of Facade Triangles 68

6 Disparity Map Fusion for 3D Model Generation 71
6.1 Algorithm Overview 71
6.2 Preprocessing of Depth Maps 72
6.2.1 Outlier Filtering Based on Support Images 73
6.2.2 Normal Computation using Restricted Quadtrees 73
6.2.3 Computation of local GSDs 74
6.3 Median-Based Filtering 74
6.4 Visibility Check 78
6.5 Results and Discussion 79
6.5.1 Fountain 79
6.5.2 Middlebury Benchmark 80
6.5.3 Airborne Oblique Dataset 81
6.6 Mesh Extraction 82

7 Summary and Outlook 89
7.1 Summary 89
7.2 Limitations and Outlook 90

8 Appendix 93
8.1 Parametric Matching Costs 93
8.2 Image Rectification 95
8.2.1 Rectification Based on Homographies 95
8.2.2 Polar Rectification 98
8.3 Additional Material for Evaluation of Depth Maps 101
8.4 Additional Material for Evaluation Fuion of Depth Maps 103

Chapter 1

Introduction

1.1 Motivation

Derivation of 3D information of objects and scenes from imagery has been, and still is, a vivid research topic in photogrammetry and computer vision. Driven by advances in technology of digital camera systems and algorithms, limits of automatic 3D surface reconstruction were pushed regarding precision, robustness, processing speed and scale in the recent years. Applications are various and range from airborne mapping using high quality imaging devices to close range reconstructions utilizing mobile phone cameras. Such reconstructions are interesting not only for measuring and documentation applications, but also for visualization and modeling purposes and as a source for scene interpretation for example in robotics or automotive driver assistance systems. In this work we mainly focus on scene reconstructions from aerial imagery. In the domain of airborne data collection LiDAR was the predominant technique for a long time. However, density and precision of surfaces which can be obtained by image driven approaches in the meanwhile are true alternatives for many applications. This success is also due to the ease of data acquisition and the flexible use of imaging sensors. Exemplary, additional to the utilization of more traditional mid- and large frame camera systems, mapping using unmanned aerial vehicles (UAVs) equipped with consumer grade cameras became popular in recent years. These platforms allow for rapid flight missions and data collection at comparable low costs. The immense diversity with respect to noise levels of sensors, image network configurations, availability of additional sensor data and the constitution of the captured structure demands for flexible and robust processing strategies within reconstruction pipelines. Many algorithms for dense reconstruction target specific applications or rely on certain assumptions or scene priors. The capability to reconstruct geometry from different cameras and network configurations possessing differing characteristics without any scene pre-knowledge is one of the key challenges to be tackled within this thesis.

Computational complexity, in particular for dense matching approaches, are tremendous and many existing methods require significant amount of time for geometry extraction. With respect to time critical applications, as for example mapping of disaster regions, run-time performance and therefore optimization is essential. On the other hand, memory requirements of these approaches often are considerable which hinders processing if memory resources are limited, for example on mobile devices. This becomes even more critical when utilizing imagery from photogrammetric camera systems possessing large dimensions or oblique images possessing large fields of depth. Regarding time and memory issues the Semi Global Matching (SGM) favorably compares to other stereo algorithms and therefore builds the core of the proposed pipeline. However, we show that by modification of the base-line algorithm memory and time demands can be drastically reduced when using priors derived by matching of low resolution versions of the images. Beside this adaption several other possibilities to reduce hardware resources will be addressed throughout the pipeline, which eventually enables our programs to run on standard hardware within reasonable times for arbitrary scene geometry.

Dependent on platforms and sensors, typical projects aim at reconstructions of cities but can span whole countries. This often results in a vast amount of data to be processed and demands for unlimited scalability of employed methods. Despite all optimizations, if datasets exceed a critical size not all data can be kept in main memory and the problem has to be divided into multiple chunks. Although dense stereo approaches naturally divide the reconstruction into many subproblems, proper handling of intermediate data in subsequent processing steps is of great importance to guarantee scalability with respect to time, memory and disk storage. We address this issue by the design of efficient tiling schemes.

Reconstructions based on airborne imagery are used for the generation of mapping or cartographic products as digital elevation models (DEM), digital terrain models (DTM), (true-) ortho photos and 3D city models. Since these derivatives rely on detailed geometry, one of the key concerns of course is the quality of reconstructions. Beside high accuracy and completeness also low amount of outliers are desirable. Within image based-reconstruction methods quality is typically improved by exploiting redundancy within the collected data. Dependent on the utilized algorithm this is performed directly in the matching stage or in case of depth map based algorithms within the fusion of depth maps. We tackle this problem by a two step approach: we match each image against several neighboring views and refine single depths based on geometric consistency and minimization of reprojection errors. Then, depth map fusion is carried out. At the time when this thesis was started the precision of airborne reconstructions derived by semi global optimization was rather unclear. To properly plan surveys regarding system specifications, flying heights, processing times and block configurations predictions regarding the achievable quality is mandatory. Therefore we rigorously evaluate our methods on well established benchmarks targeting the generation of digital surface models as well as close range reconstruction.

Flight patterns for data acquisition are highly dependent on the desired products. Whereas nadir patterns typically serve as basis for the generation of DSMs, DTMs and true-ortho photos, oblique camera systems are used if real 3D structure should be extracted which is useful for the generation of 3D city models or facade analysis. Whereas in the latter case extraction of 3D structure is explicitly desired, in the first case the presence of 3D information leads to artifacts and must be filtered. This in particular holds true if wide angle lenses are utilized or even more critical, DSM are generated from oblique image imagery. Therefore we propose a depth map fusion approach to generate 2.5D models (DSMs) which improves existing approaches with respect to reliable geometry in scene regions where 3D structure is extracted, for example at building edges. For the fusion of depth maps aiming at reconstruction of 3D scenes other problems arise. Observations in single depth maps in general possess large variances in ground sampling distances and precisions. Latter is due to fronto-parallel effects, differences in redundancy, properties of ray intersections, variances in GSDs, inaccurate orientations, blurred image regions etc. We argue that the combination of all these effects is hard to model. In order to account for outliers we propose a novel median based approach for 3D depth map fusion.

Meshed surface representations are widely used for visualization purposes. Moreover, since directly encoding neighborhood information they are increasingly used in subsequent algorithms as for example for interpretation of 3D data. To meet this requirement we propose methods for mesh generation based on the generated DSMs and 3D fusion results.

.

1.2 Objectives

The objective of this thesis is to build and evaluate a flexible system for dense surface reconstruction giving possibly precise and blunder free surfaces. The input is a set of images along camera poses, the output is, depending of the application, 2.5D structure represented as gridded elevation data or 3D structure stored as point clouds or triangle meshes. Thereby following capabilities are of particular importance:

- Scalability: Since real world data sets often consist of blocks of thousands of images the algorithm should scale well to resolution and amount of images. Thereby processing should not be restricted to specialized high-end hardware clusters but should also be possible on standard desktop computers.
- Scene independence: Developed methods should work for 2.5D and 3D scenes. In the proposed approach image matching and multi-view triangulation is handled by exactly the same algorithm. How-

ever, approaches for the fusion of resulting depth maps differ because of different characteristics of data sets. Whereas data collection schemes for typical aerial applications aim for 2.5D surface reconstructions, in most close range datasets the extraction of 3D structure is required. Therefore two different strategies for depth map fusion are implemented.

- Precision and outliers: Redundancy across depth maps should be exploited to increase precision of surfaces and eliminate blunders. Accuracy of stereo matching is dependent on many parameters such as radiometric quality, signal-to-noise ratios, distinctiveness of texture, image blur, deviation of assumptions regarding fronto parallelism, scale variances across image pairs etc. Moreover, precision of triangulated 3D information is dependent on image scale and ray intersection angles within stereo or multi-view configurations. Theses properties are largely varying across views in many image sets and therefore have to be handled in a proper way.

- Processing speed: Due to large amount of data processing should be fast. Efficient formulations of problems and data structures, as well as parallizable design of algorithms are key requirements.

- Automatic adaption of parameters: In order to minimize user interaction and enable non-expert use the process should be robust to parametrization and if necessary adapt parameters automatically.

Matching quality is highly dependent on image similarity. Thus, for small base line configurations matching works well in general. In contrast, ray intersection angles are small and precision of forward intersection is limited. Therefore a proper strategy for the selection of image pairs to be incorporated into reconstruction process has to be investigated, in particular for high redundant data sets. The discussed algorithms are evaluated regarding processing time, memory demands, precision and completeness. This includes comparisons to existing algorithms implemented by academia as well as commercial reconstruction pipelines.

1.3 Main Contributions

The first contribution of this thesis is an efficient coarse-to-fine adaption of the SGM algorithm dynamically identifying homologous regions across image pairs and adapting disparity search ranges according to surface priors. Compared to the original method this strategy allows for improved processing times, reduced memory demands and resolves ambiguities within the search of correspondences, which results in improved depth maps for challenging texture. The second contribution is a framework for multi-view forward intersection scaling well to large data sets. It consists of an outlier rejection scheme based on geometric consistency and forward intersection based on minimization of the reprojection error. Both parts consider geometric properties of camera configurations as varying ray intersection angles and differing image scales across views. Working on epipolar imagery the multi-view intersection problem can be formulated in a closed form for different types of rectification geometries which avoids costly matrix inversions in the course of solving linear systems or iterative approaches. Furthermore, we present a simple but efficient method for the computation of 2.5D elevation data. This method scales well to large amount of data (blocks of thousands of images) and improves existing methods by preservation of abrupt height jumps whilst small details are not filtered. Whereas crack-free mesh extraction of such elevation data was covered in multiple works (e.g. [Pajarola et al., 2002], [Pajarola, 1998]), we enhance the method of restricted quadtree triangulation by a re-meshing procedure at depth discontinuities whilst maintaining unlimited scalability. Additionally we present a novel method for the fusion of depth maps representing real 3D structure. Based on geometry- adaptive meshes extracted from the single depth maps, robust normals are reconstructed. These normals define the main filter direction within a median-based fusion framework. Latter utilizes oc- and kd-trees and accounts for varying resolutions within datasets. We show that the resulting oriented points can be used to produce high-quality meshes.

1.4 Outline

This document is separated in five main parts as follows. Consecutively to the introduction we start with a review of relevant related work in the field of dense surface reconstruction to classify the implemented pipeline. This comprises structure of MVS reconstruction workflows, dense stereo and multi-view matching approaches. Furthermore, related work on depth map fusion will be discussed. The main part of this document will discuss the developed algorithms and their evaluation. This involves conceptional ideas as well as implementation details. After introducing the outline of the reconstruction pipeline in chapter 3 the algorithm sections can be subdivided into three main parts:

- Disparity map computation: Since the proposed stereo matching algorithm is based on epipolar images, in section 4.1 different types of image rectifications are reviewed. We put emphasis on this since the subsequent 3D point triangulation is based on the established epipolar constraints. Furthermore, in section 4.2 the actual dense matching algorithm is discussed.

- Multi-View Depth Maps: Each image is matched against multiple proximate images to generate redundant depth observation. Redundant observations are linked and evaluated regarding geometric consistency. Besides exploiting this redundancy to eliminate outliers, multi-view triangulation is utilized to improve precision of depths. In section 4.3 we explain our approach for forward intersection minimizing reprojection errors.

- Depth Map Fusion: In these chapters the generation of the actual surface models will be explained. In order to generate consistent surfaces the depth maps derived from the implemented MVS are fused. First, in chapter 5 the implemented approach for depth map fusion of 2.5D models (DSMs) is explained. Then, the fusion method for 3D surfaces is discussed in chapter 6. Both sections are concluded by methods for mesh generation.

Each of these chapters includes an evaluation part identifying advantages and limitations of proposed algorithms as well as comparisons to other commercial and academic MVS pipelines. The thesis is concluded by a chapter summarizing the findings and limitations and giving an outlook for future work and possible improvements.

Chapter 2

Related Work

2.1 Multi-View Systems

In the last three decades tremendous amount of work in the area of multi-view systems as well dense stereo correspondence was conducted. To classify and establish similarities as well as differences of the proposed methods and other reconstruction systems we first summarize the most important concepts and characteristics of MVS systems. Following the taxonomy of [Seitz et al., 2006] MVS can be classified using certain criteria describing the most important properties of reconstruction systems as summarized in the following sections. The type of *reconstruction algorithm* describes the general strategy of reconstructing surfaces as discussed in section 2.1.1. In the course of the reconstruction process the surface state have to be represented by some data structure or implicit or explicit function, most common types are explained in section 2.1.1 *Scene representations*. One of the key components is the similarity measure used to establish pixel correspondences across views. Tremendous amount of algorithms have been proposed in the photogrammetric and computer vision community, those closely related to dense reconstruction are highlighted in section 2.1.3. In almost every state-of-the-art MVS framework shape priors are utilized. They support reconstruction in areas where image similarity is not distinctive by assuming smoothness of the surface. Simultaneous optimization of photo consistency and shape priors can be formulated as energy minimization problem and to large degree depends on the representation of the scene. Common concepts of energy minimization in MVS and stereo are reviewed in section 2.1.5. Because the proposed MVS system is based on dense stereo, we review relevant concepts in section 2.2. Since large parts of methods as photo consistency, scene representation and energy minimization are similar to those in MVS, we focus on the problem formulation, refinement of disparity maps and filter techniques. The chapter is closed by a review of related work in the area of depth map fusion and common concepts for surface extraction from point clouds. As in nearly all work conducted in MVS, we assume that is exterior and interior camera parameters are known. The reader interested in computation of camera orientation is referred to [Triggs et al., 2000], [Agarwal et al., 2010], [Snavely et al., 2006], [Agarwal et al., 2009], [Wu, 2011] as a starting point to the subject and examples of state-of-the-art methodology.

2.1.1 Reconstruction Algorithms

According to [Seitz et al., 2006] MVS systems can be categorized by four types of reconstruction algorithms. The first class typically operates on volumes (e.g. octrees, multilevel octrees). For each entity of the volume (e.g cube) a cost representing the probability of the respective entity being part of the surface is assigned. Based on the assigned costs a surface is extracted. Exemplary algorithms are voxel coloring [Seitz and Dyer, 1999] and frameworks based on Markov Random Fields (MRF) where surfaces are modeled as 2.5D elevation maps [Pierrot-Deseilligny and Paparoditis, 2006], [Vogiatzis et al., 2008]. In contrast, the second type of methods start with coarse representation of the surface which then is iteratively refined minimizing an energy functional. Typical representatives include level sets [Pons et al., 2007] and mesh evolution algorithms [Hiep et al., 2009]. Patch-based algorithms like [Furukawa and Ponce, 2010] start with high confident surface points and grow the surface utilizing geometric information of the points already reconstructed. Space carving [Kutulakos and Seitz, 1998] starts with a solid and iteratively carves the volumetric entities not being photo-consistent. The third type of algorithms are depth map based. Geometry is reconstructed for single images by stereo or multi-baseline stereo (e.g. [Okutomi and Kanade, 1993]). In a final step reconstructed depth maps are fused. Examples include the approaches proposed in [Merrell et al., 2007], [Goesele et al., 2007], [Pollefeys et al., 1998]. The last type of algorithms reconstruct point sets by feature matching to which eventually a surface is fitted.

2.1.2 Scene Representations

Scene representation describes the mathematical framework or data structures which are used to represent the extracted surface. Most common types are triangle or polygon meshes, elevation maps, depth maps or voxel representations based on occupancy information or level sets. Meshes are sets of planar faces sharing edges to form a surface. Beside low memory requirements, this representation allows for accelerated processing since many operations suit computation on GPU hardware. Example for mesh based methods were proposed by [Hiep et al., 2009] and [Yu et al., 2007]. Elevation maps, more commonly referred to digital surface or terrain models in the photogrammetric community, map values $f : \mathbb{R}^2 \mapsto \mathbb{R}$ from a discrete parameter domain $\Omega \subset \mathbb{R}^2$ to elevation values representing the surface $S = f(\Omega)$. This representation is convenient to handle and adequate whenever it is sufficient to reconstruct 2.5D structure for example for reliefs [Vogiatzis et al., 2008] and surface models generated from airborne cameras [Pierrot-Deseilligny and Paparoditis, 2006], [Bethmann and Luhmann, 2015], or more generally, for reconstructions for which data is collected from similar viewing directions. Many methods use depth maps to store 3D information of the scene. This is an obvious concept because large amount of algorithms are based on stereo or multi-baseline stereo directly producing disparity or depth maps. However, to extract consistent non-redundant surfaces the fusion of all depth maps is necessary which is challenging since single depth maps are typically reconstructed possessing variances in depth precision. Depth map based approaches scale very well to large scale data sets since splitting the reconstruction problem in many sub-problems by design. Another popular scene representation is based on voxels utilized by volumetric reconstruction algorithms [Slabaugh et al., 2001]. Thereby the region subject to the reconstruction is discretized in cubic volumes and spatial topology mapped by octrees. One possibility of representing the surface is to assign occupancy information to each voxel as done in Visual Hulls, Voxel Coloring [Seitz and Dyer, 1999] and Space Carving. Another possible voxel representation is based on level set theory developed by [Osher and Sethian, 1988] and first utilized for surface reconstruction by [Faugeras and Keriven, 1998]. In that approach the surface is represented as implicit function $f : \mathbb{R}^4 \mapsto \mathbb{R}$ where the surface is given by $f(x, y, z, t) = 0$. Thereby x, y, z denote spatial coordinates of the volume and the parameter t is denotes the state of the surface at a given time or iteration. However, memory requirements of voxel based approaches are significant even when utilizing multi-level data structures.

2.1.3 Photo Consistency Measures

Photo consistency is the similarity measure representing how well intensities between corresponding image pixels or patches match. Generally matching of correspondences can be divided in two main categories: *feature-based* and *area-based.* *Feature-based* similarity measures are extracted at salient positions in the images representing edges, corners or points possessing distinctive characteristics. Characteristics of neighboring intensities around the extracted coordinates are decoded by descriptors and correspondences across views can be derived by descriptor comparison. One of the most popular feature due to robustness against scale, illumination and rotation is SIFT [Lowe, 1999]. An GPU implementation of the algorithm was proposed in [Wu, 2007], an adaption adding invariance to affine distortions was published in [Morel and Yu, 2009]. *Features-based* techniques are mainly used in algorithms dealing with sparse correspondences and limited initial knowledge of the scene, thus making the application of shape priors difficult. Popular examples are structure-from-motion methods like [Verhoeven, 2011], [Snavely et al., 2006] [Wu, 2011]. However, *feature-based* matching is sometimes utilized in early stages of surface reconstruction frameworks to initialize a surface which is then subject to further refinement [Hiep et al., 2009], [Labatut et al., 2007], [Furukawa and Ponce, 2010]. For dense matching applications *area-based* similarity measures are of more relevance due to reduced computational effort and better subpixel accuracy. The problem of limited robustness is typically tackled by implying shape priors and search space limitations derived by hierarchical processing schemes. Generally we distinguish *parametric* and *non-parametric* matching costs [Hirschmüller and Scharstein, 2007]. *Parametric costs* are computed based on the intensity values. More precisely, based on the intensity difference of two pixels (and the intensity differences of their neighbours located in a rectangular windows) the goodness of the match is defined. The most common parametric cost are sum of absolute differences (SAD), sum of squared differences (SSD) and its zero-mean versions ZSAD and ZSSD as well as normalized cross correlation (NCC). Formulas of the single matching costs are given in section 8.1. ZSAD, ZSSD and ZNCC compensate for constant offsets. ZNCC additionally compensates for gain. An iterative procedure called least squares matching (LSM) based on SSD was proposed by [Gruen, 1985]. Using rectangular windows for the computation of matching costs implies that surfaces are fronto-parallel. Since this assumption is violated for almost every real world scene the authors formulate a matching cost minimizing the squared sum of differences subject to an affine transformation of the image patch. The concept can be enhanced for multi-photo consistency. The methods proposed in [Baltsavias, 1991] [Gruen and Baltsavias, 1988] take this approach further by additionally restricting the search spaces using the known interior and exterior orientation. Geometric and radiometric constraints are formulated in one non-linear system which can be linearized and solved iteratively. These costs have been successfully used in various reconstruction systems, exemplary [Grun and Zhang, 2002] and [Goesele et al., 2007].

The most popular *non-parametric* matching costs comprise rank transform, soft-rank transform and the Census correlation matching costs [Zabih and Woodfill, 1994]. Beside intensity values in windows around a reference pixel also their spatial location is encoded. The rank transform and soft rank transform are operators applied to images before computation of an AD matching score. The rank operator T compares the intensity of a reference pixel i_r to a set of intensities at neighboring pixels i_n and evaluating to 1 if $i_n <= i_r$ and to 0 otherwise. The results of all comparisons are summed up. The soft rank operator eases the sensitivity to noise by reducing the impact of intensity differences in the noise band. The census transform generates a binary string for a pixel using the rank operator T. The final matching costs of two pixels is then derived by computation of the Hemming distance. Since not directly operating on intensity values these matching costs are robust with respect to radiometric variances across images.[Sun et al., 2011] reported good results by combining AD and census matching costs.

Mutual information (MI)[Viola and Wells III, 1997] was used as pixel-wise matching cost in the publication [Hirschmüller, 2008]. It is based on statistical information how well intensities of two images are aligned. More precisely, MI is based on the entropies of the cross-probability distribution of two overlapping image parts and the entropies of probabilities of the single images. For well aligned images the cross probability histograms posses few distinct peaks, whereas for non-aligned images the histograms are flat. Practically initial probability histograms are computed by defining random pixel correspondences. Based on that entropies and cross entropies image pairs are matched and the results are utilized to update the probability histograms. Since the matching cost is truly pixel-wise good results at discontinuities can be obtained. However, the entropies are dependent on the image size and content which complicates a robust parametrization.

[Tola et al., 2008] proposed the DAISY descriptor for wide-baseline stereo. Inspired by SIFT, descriptors are build for each pixel using gradient orientation histograms and matched across the images. The approach yields excellent results, at the cost of processing speed and memory requirements.

All previous matching costs are based on the assumption of diffuse or Lambertian reflectance of surfaces, which is violated for almost all real-world scenarios. This holds particularly true for challenging surface materials as glass or plastics. Several works have tackled reconstruction for non-Lambertian surfaces. For example

[Yang et al., 2003] model the reflectance as a sum of specular and diffuse reflectance. Correspondences of diffuse surface parts possess distributions around a single point in color space. In contrast correspondences of specular reflectance are supposed to form lines from object color to color of the ambient light in color space. By analyzing color variances of correspondences and checking line and point hypotheses the probability of the surface type is derived and matching costs are computed accordingly. [Yu et al., 2007] use View Independent Reflection Maps (VIRM) to to model non-Lambertian surfaces. They design their reconstruction algorithm as a interleaved optimization, one part optimizing the VRIM the other optimizing object shape based on image similarity. Instead of defining matching costs modeling the specular reflectance [Sinha et al., 2012] use two layered depth maps constructed by dense stereo to recover reflective and diffuse components. Matching costs are analyzed for two peaks along epipolar lines identifying regions where actual surfaces are covered by non-Lambertian surfaces. Based on the foreground and background disparities planes are fitted and clustered. Eventually each pixel is labeled as two layered or single layered region using binary labeling based utilizing graph cuts (α-expansion) and two layered peaks are assigned to two depths according to the plane hypotheses.

2.1.4 Visibility Models

Most state-of-the-art reconstruction pipelines utilize visibility models to account for physically impossible surface states. The most elegant approach is to use the current surface representation to compute visibility as done in many surface growing algorithms [Hiep et al., 2009] [Faugeras and Keriven, 1998]. Beside this *geometric approach*, *quasi-geometric* approaches try to limit effects of occlusions by restricting views to be matched to nearby camera stations possessing similar viewing directions [Goesele et al., 2007], [Goesele et al., 2006]. Due to the restricted changes in viewing directions also the number occlusions is reduced. Typically heuristics are based on angles of viewing directions in combination with the length of baselines. [Furukawa et al., 2010] proposed a method for multi-view clustering which, beside dividing large datasets into smaller sub-problems, selects views to be matched to reduce redundancy and to assure completeness. They cast this problem into a energy minimization problem. The last technique is referred to as *outlier-based* for which occlusions are linked to outliers. Detection is implemented by heuristics based on geometric consistency or consistency of image similarity measures across multiple views. Exemplary [Koch et al., 1998] track correspondences across depth maps and non-visible surfaces are detected due to their violation of geometric consistency. [Drouin et al., 2005] utilize per-pixel visibility masks which encode in which views a pixel is observed. Most probable visibility configurations are computed based on photo consistency of different visibility combinations. Iteratively an energy functional optimizing surface smoothness and image similarity are computed and viability masks are updated. To guarantee convergence entities in visibility masks are solely deactivated and never activated. [Goesele et al., 2007] use pairwise computed NCC scores and views providing a low score indicate non-visible surface parts in stereo models (beside non-beneficial radiometry, viewing angles, etc.) which then are excluded from further processing.

2.1.5 Shape Priors and Optimization Concepts

To identify homologous image points across imagery some sort of image similarity measure is utilized (see section 2.1.3). However, in areas of repetitive or week texture or non-Lambertian surfaces, costs loose their distinctiveness. Shape priors model assumptions, typically the smoothness of the reconstructed surface, to enforce accurate and consistent reconstructions for challenging surface areas. Almost all modern reconstruction algorithms are stated as some sort of energy optimization problem, minimizing an data driven energy term E_{data} composed of matching costs and a term incorporating shape priors E_{prior}. Utilizing an optimization framework, disparities are estimated such that an overall energy of the form

$$E = E_{data} + \lambda E_{prior}. \tag{2.1}$$

is minimized. Of course the formulation of the energy to be minimized and the attendant is highly dependent of the surface representation. In the following we first review the work related to variational optimization and tetrahedralization. Then, more closely related to our work, we highlight common optimization concepts in the area of dense stereo.

Level-set reconstruction algorithms [Faugeras and Keriven, 1998], [Pons et al., 2007] are based on variational optimization. Thereby a cost function is defined integrating image similarity, surface smoothness and visibility constraints over an initial surface represented by $f(x, y, t)$. The problem now is to find an evolved surface function $f(x, y, t+1)$ minimizing the cost integral. The minimizer of the cost integral is given by the Euler-Lagrange equation, a second order partial differential equation (PDE). By iteratively solving the PDE, an update of the surface function can be obtained. Starting with the coarse guess, the surface is evolved in a way that the functional f converges to a solution minimizing the cost integral. A nice property of level set algorithms is that topology of the surface can change during evolution such that the guess of the initial surfaces can be rather coarse. A mayor draw back is that level set methods do not scale well to large datasets.

Another variational algorithm yielding impressive results on aerial as well as close range data sets was proposed by [Hiep et al., 2009],[Vu, 2011]. Because representing the scene as triangle mesh from an early point in the reconstruction process, it scales well to large scale datasets. Actually the algorithm is a two step approach, first constructing the surface mesh close to the actual surface based on tetrahedralization and then a mesh refinement using variational optimization. Thereby vertex positions are iteratively updated (using the Gauss Newton method) such that the sum of pair-wise image consistency costs as well as the thin plate energy [Kobbelt et al., 1998] (a measure of bending) are minimized. Eventually the mesh converges to an optimal state. A crucial point in the course of energy minimization is the computation of gradients of the energy functional defined as a function of the mesh [Delaunoy et al., 2008]. It is mentioned that the method relies on a good initial guesses for good convergence. Furthermore, this is important due to the fact that the topology the mesh can not change in the course of mesh optimization.

[Labatut et al., 2007] utilize the concept of tetrahedralization. Based on a set of homologous points derived by feature-based matching the authors construct a Delaunay triangulation after linking corresponding points forcing geometric consistency of pairwise matches. 3D coordinates of consistent observations are obtained by minimization of reprojection errors. The dual of the resulting Delaunay triangulation divides the space in tetrahedra which are then subject to a binary inside-outside labeling. This labeling is retrieved via graph cuts (s-t cuts) minimizing an energy functional based on visibility of vertices, photo consistency and minimization of the surface area (surface smoothness). These costs are represented by edges in the graph and each tetrahedron is interpreted as vertex additionally connected to the sink and source node. The resulting minimal cut, the faces between the tetrahedra labeled as outside and those labeled as inside, define the final surface. [Jancosek and Pajdla, 2011] enhanced the framework for weakly supported surfaces yielding improved results in areas of sparse scene coverage.

Due to the structured topology, representing the surface as 2.5D elevation data or depth map as in dense stereo allows for more convenient formulation of energy optimization. A popular framework of modeling the correspondence problem with respect to equation 2.1 are Markov Random Fields (MRF). A MRF is an undirected graph where each node represents a random variable. Spatially neighboring nodes possess conditional relationships whereas non-neighboring nodes do not influence each other. In case of stereo matching each pixel is modeled as a node with a non-observable hidden variable, the disparity, and a observable variable based on the intensity values. The assumption that neighboring pixels posses similar disparities is represented by the conditional relationships between proximate nodes such that smoothness of neighboring disparities can be forced. The set of disparities, also called labels, minimizing the energy functional can be derived using different strategies. Most common ones, due to acceptable trade-off between computational efficiency and accuracy, are Loopy Believe Propagation and Graph Cut algorithms. Loopy Believe Propagation (e.g [Freeman and Pasztor, 1999], [Sun et al., 2003], [Felzenszwalb and Huttenlocher, 2004]) iteratively computes the likelihood of candidate disparities of each node. Note, that in dense stereo the problem is modeled as undirected graph and no guarantee for convergence to the global minimum is given. For each possible label of each pixel a likelihood is computed based on the image similarity and beliefs about the nodes disparity propagated from neighboring nodes. The propagated likelihoods, also called a message, are weighted by a smoothness term, penalizing variances of neighboring hidden variables. Iteratively messages are computed and then passed to the neighboring nodes which use this information for the next message update. [Tappen and Freeman, 2003] report two different strategies for the message update schedule. The synchronous update scheme computes messages for each node. On completion the messages are passed and the message update takes place. A second strategy, referred to as the accelerated processing scheme, possesses faster convergence due to faster propagation. It is based on updating and passing messages in one image direction, for example right to left, using messages from left, upper and lower nodes from the previous iteration. In the same manner messages are passed from right to left, up to down and down to up. It is desirable that in regions of depth discontinuities less smoothing is applied than in regions of smooth surfaces. Assuming that depth discontinuities occur at large gradients of intensity images, [Tao et al., 2001] tackle this problem by segmentation of intensity images and constructing one MRF per segment such that disparity optimization of different segment becomes decoupled. More elegant, [Sun et al., 2003] model limited smoothing at discontinuities directly in their probability function.

A second strategy to find labels solving the MRF problem are Graph Cuts algorithms. The general outline proposed in [Boykov et al., 2001] is finding the optimal labeling within the set of all possible labels by sequentially optimizing one or two labels such that the energy in equation 2.1 is minimized. An pairwise optimization of two labels α and β is called α-β swap, the optimization of a single label α is called α-expansion. One iteration comprises α-expansion over all labels, or α-β swaps of all label combinations. Several cycles of iterations will minimize the global energy. The iterative approach terminates if the energy remains constant. The crucial point of the algorithm are α-β swap and α expansion label optimizations which are based on Graph Cuts. To grasp the idea of the algorithm the α-β swap strategy is explained in more detail, however α-expansion follows a similar concept. A undirected graph is constructed as follows: Each hidden variable (pixel) is modeled as node. Furthermore, two terminal nodes are introduced representing the two potential labels α and β. Directly neighboring nodes are connected by an edge, so called n-links if they are already labeled α or β. Moreover, each node is connected by an edge to a terminal node, so called t-links. The weights of terminal links are composed of the data term (image similarity measure) and a smoothness term penalizing label variances between the node itself and surrounding nodes. The edge weights of n-links apply a penalty if neighboring nodes do not provide similar labellings α or β. The edges of the assembled graph are now cut such that the sum of cut edges are minimal, also called a minimal cut. This cut defines the updated labeling leaving initial labels unchanged or relabeled as α or β minimizing the energy functional. Additional to image similarity and label smoothness [Kolmogorov and Zabih, 2001] construct a graph enhanced by an additional energy for occlusions and compute a minimum via graph cuts.

Another graph optimization method for stereo and multi-view matching modeled as MRF problem was proposed by [Roy and Cox, 1998]. It is based on the max-flow min-cut algorithm. Thereby the matching problem is modeled as a 3D graph similar to the belief propagation algorithm. Each pixel with a potential disparity (x, y, d) is modeled as node. It is connected by four edges (occlusion edges) to its neighbours at constant disparity. Additionally two edges connect the node (x, y, d) to $(x, y, d + 1)$ and $(x, y, d - 1)$. The layer containing nodes with maximum disparity (x_i, y_j, d_{max}) i, j are connected to a sink node whereas nodes located in the lowest layer (x_i, y_j, d_{min}) are connected to a source node. The weights of disparity edges $c_d(u, v)$ connecting node u and v are computed as the average of the image similarity costs of two d-connected nodes (x, y, d), $(x, y, d+1)$ and (x, y, d),$(x, y, d - 1)$ respectively. The weight of occlusion edges controls the smoothness of extracted surfaces and are computed as $kc(u, v)$ with the smoothness parameter k. The maximum flow from source to sink can be derived using typical algorithms from graph theory, for example [Boykov and Kolmogorov, 2004], [Goldberg and Rao, 1997], [Cormen et al., 2001]. Based on the maximum flow, the min cut can be derived. The cut can also be interpreted as the bottleneck limiting the flow from source to sink in the graph. It represents the set of nodes defining the disparity surface minimizing the functional of the form 2.1. [Pierrot-Deseilligny and Paparoditis, 2006] presented an coarse-to-fine adaption of the algorithm easing memory demands and computational complexity which performed well in benchmarks as [Haala, 2013b]. [Ishikawa, 2003] showed that, for shape prior energies which are formulated as convex functions and meaningful linear labeling ordering, the labeling problem can be globally solved in polynomial time. [Pock et al., 2010] transfer the concept from discrete to contentious space where a global optimal solution can be derived using variational optimization.

One of the first algorithms for stereo correspondence using energy minimization incorporating photo-consistency and shape prior were based on dynamic programming (for example [Baker and Binford, 1981], [Ohta and Kanade, 1985]) matching pixels possessing salient gradients. Dynamic programming in stereo algorithms basically computes the disparities along a 1D scanline (epipolar lines) in a way such that an energy functional of the form 2.1 is minimized. Thereby the minimization problem is split into sub-problems which are solved and the solutions are used for the minimization of the next larger problem in a recursive manner. [Geiger et al., 1995], [Belhumeur, 1996], [Cox et al., 1996] proposed dynamic programming using probabilistic frameworks to match correspondences pixel-wise. The idea of dynamic programming is utilized in the SGM algorithm which is discussed in depth in section 4.2. The main problem of the concept is that results suffer from streaking effects since optimization including priors is only handled for single scan lines and couplings between scanlines are not considered in the course of optimization.

The SGM algorithm [Hirschmüller et al., 2012], [Hirschmüller, 2008] is an extension of the previously described scanline optimization. In contrast to dynamic programming approaches costs are accumulated along several scanlines instead just of one. Despite increasing computation time this significantly reduces streaking effects. The algorithm has gained a lot of attention in the recent years. Due to fast processing times at acceptable precision and robustness to parametrization it is often the choice of stereo algorithm in real world automotive, close-range and mapping applications. In particular the ability to maintain sharp depth discontinuities at object boundaries adds great benefit to many applications. The algorithm can be scheduled in parallel which allows for efficient implementations on GPUs [Rosenberg et al., 2006], [Gibson and Marques, 2008], [Ernst and Hirschmüller, 2008] and FPGAs [Gehrig et al., 2009], [Banz et al., 2010] which makes it suitable for real-time applications. Since the algorithm is the basis of our reconstruction system it will be reviewed in detail in section 4.2. [Hermann and Klette, 2012a], [Hermann and Klette, 2012b] independently to this work proposed an hierarchical approach for the SGM algorithm for automotive applications. They initialize full resolution correspondence search by utilizing priors of matching half resolution imagery. Thereby the disparity search range for full resolution processing is narrowed down pixel-wise. They point out that initialization of search ranges by propagation of the priors is a crucial point. Their algorithm performs rather well on the KITTI benchmark datasets [Geiger et al., 2012]. However, a method for data fusion for multiple depth maps is not addressed. A memory effective flavor of the SGM algorithm was proposed by [Hirschmüller et al., 2012]. Instead of storing aggregated costs of all disparity candidates, which implies allocation of structures with the dimensions $rows \times cols$ and a constant disparity range, only the minimal costs induced by each of 8 paths are considered. The author argue that is unlikely that the final disparities are located at other positions than that of one of the path minima. Although memory demands are reduced significantly, in particular for large disparity ranges, cost aggregation requires three passes (instead of two for the classic method) which results in an increase of processing time. However, the algorithm enables fast processing of scenes possessing large disparities on hardware as GPUs and FPGs on which memory is limited and memory band-with plays a crucial role.

2.1.6 MVS with Regard to the Generation of Elevation Data

The automatic generation of elevation data or DSMs is a long studied problem, particularly in the photogrammetric community. These algorithms can be seen as sub-class of MVS and since closely related to our methods, important work in this field is reviewed in this section. Building on the basic concepts, as image similarity and energy optimization which were discussed in previous parts, we discuss some exemplary workflows for the generation of 2.5D elevation data. Most early works build up on feature based matching and subsequent densification of the extracted feature points [Krzystek, 1991] [Newton, 1999] [Grun and Zhang, 2002] [Maas, 1996]. [Newton, 1999] construct a Delauney triangulation based on points derived by feature matching. The triangulated irregular network (TIN) is then used to initialize area based matching of grid points. In regions where matching fails a fall-back strategy is applied performing matching on spatially nearest grid points serving as better initial guess for LSM matching. A major problem is the matching process itself due to non-distinct or varying texture across the images. Reasons comprise texture-less areas, non-Lambertian effects, moving shadows, occlusions and repetitive structures. These difficulties can be eased by reducing ambiguities in the course of correspondence search. This can be realized by limitation of search spaces for possible matches using coarse-to-fine strategies. [Krzystek, 1991] construct DTMs using image pyramids from which features are extracted and matched across images along epipolar lines. The so derived points are subject to a finite element method fitting a surface such that points not element of the actual terrain are removed. Extracted surfaces are used to initialize matching on higher pyramid levels. Similarly, an algorithm designed for TLS (Three Line Scanner) data utilize a hierarchical processing scheme. Thereby feature points are extracted in a reference view and transfered to the search images using exterior orientation and average terrain heights. Based on the windows around correspondences cross correlation scores are computed and thresholding define valid matches. The initial surface is used for search space limitation in a second matching stage. To further reduce ambiguities in matching, the author proposes to force shape priors using discrete relaxation as described in [Hancock and Kittler, 1990]. However, in a final step the surface is refined using MPGC and GCMM neglecting any assumptions of local smoothness. As mentioned before, the concept of forcing shape priors is an essential mechanism to resolve ambiguities of the correspondence problem in particular for challenging texture. [Pierrot-Deseilligny and Paparoditis, 2006] combine hierarchical processing and simultaneous optimization of multi-photo consistency and surface smoothness. Therefore they cast the problem of elevation computation into a MRF problem which is solved using graph-cuts (MinCut-MaxFlow, see section 2.1.5). Approaches also based on MRF were proposed by [Bethmann and Luhmann, 2015] and [Irschara et al., 2012]. The first algorithm implement energy minimization by SGM, the second method utilizes the globally optimal variational algorithm based on [Pock et al., 2010]. Theses approaches reconstruct elevation data possessing dense sampling up to pixel level which leads to reduced sampling errors in particular for areas of undulating terrain. The approach proposed by [Hirschmüller et al., 2012] reconstructs depth maps utilizing stereo matching which then are merged in a subsequent fusion step. By defining a grid parallel to the surface and ortho-projection of matching results the surface is derived using median filtering. This approach is motivated by the assumption that for nadir airborne nadir configurations normals of the observed surface can be approximated by the normal of a single plane. Despite the simplicity of the fusion of matching results errors are introduced in presence of real 3D structure for example undercuts as roof overhangs.

2.2 Dense Stereo

Since the proposed algorithm is based on a dense stereo algorithm, in this section we review recent work and basic techniques of dense disparity map computation. Since the concepts of scene representation, optimization and shape priors are similar to these of MVS systems (see section 2.1.5 and 2.1.3) we focus on the main problem formulation, refinement of disparity maps and filter techniques in dense stereo algorithms. For a in-depth overview of state-of-the-art algorithms the interested reader is referred to [Hirschmüller and Scharstein, 2007] and the well known Middlebury stereo benchmark.

2.2.1 Problem Formulation

The problem of dense stereo can be stated as densely establishing the correspondences between pixels across a pair of images representing the same object point for all pixels in the images. Typically the input for dense stereo is a set of rectified images, meaning that potential correspondences across an image pair are located on the identical rows. Rectified images can be computed based on interior and exterior orientations. Various rectification approaches were investigated for example [Fusiello et al., 2000], [Loop and Zhang, 1999], [Pollefeys et al., 1999] [Abraham and Förstner, 2005]. Working on rectified imagery, the search space for correspondences is reduced to one dimension, thus the complexity of computations is eased. Correspondence of a pixel $\mathbf{x}$ in the first (base / reference /master) view and a the pixel $\mathbf{x}'$ in the second (match / search /slave) image then can be encoded by the parallax or disparity d:

$$\mathbf{x}(x, y) \leftrightarrow \mathbf{x}'(x + d, y). \tag{2.2}$$

As for MVS systems similarity measures are mainly area-based, designed for efficient computation and good sub-pixel accuracy (see section 2.1.3). For a potential pair of correspondences $\mathbf{x}$,$\mathbf{x}'$ across to images similarity measures define the matching cost $C(x, y)$. Large costs indicate that the pixels are not matching well, low costs indicate high probability of homology. Operating on rectified images for each pixel $\mathbf{x}(x, y)$ the costs $\mathbf{C}(x, y, d)$ for a disparity d in some constant range can be computed and represented as a 3 dimensional cost structure also called disparity space image (DSI). At this point it has to be mentioned that the concept of DSI can also be adapted for multi-view matching: each $\mathbf{C}(x, y, d)$ is then composed by a combination of image pair-wise similarity measures. Since pixel-wise costs might locally be not distinctive, costs are accumulated (for example using simple averring of costs) within support regions in the 3D cost structure. Thereby aggregation can be implemented for fixed disparity ranges (2d aggregation) or in three dimensions within the DSI easing fronto-parallel effects. Note that window-based similarity measures already implement 2D aggregation by design. The main point in which stereo methods differ are strategies for disparity optimization. Local approaches derive the disparity estimates directly from the aggregated costs $\mathbf{C}(x, y, d)$. Thereby it is assumed that minimum costs $\mathbf{C}(x, y, d)$ at a pixel position (x, y) indicate the correct disparity d. However, for surfaces where similarity measures are not distinctive, e.g in areas where only limited texture is provided or in case of non-Lambertian surfaces such approaches easily fail. In contrast, global methods incorporate some shape priors enforcing locally consistent depths or disparities, that is implying smoothness constraints. As for MVS this problem is typically casted as an energy minimization problem, composed of a data term and a smoothness term (see equation 2.1). Typical optimization strategies are loopy believe propagation, graph cuts, dynamic programming or semi- global matching (see section 2.1.5). From the optimized costs the most probable correspondences can be selected by winner takes it all strategy.

2.2.2 Disparity Refinement

Most stereo methods operate in discrete space meaning estimating disparities defining correspondences located at full integer pixel locations. In order to estimate floating point sub-pixel disparity fitting curves to the costs defined for every pixel is a common technique. Therefore quadratic functions are most common since sub-pixel locations can be directly computed in a non-iterative way at beneficial processing speed. However, a pixel-locking effect, a systematic error of refined disparities biased towards the integer positions, can be observed. This error is caused by a linearization error as stated in [Shimizu and Okutomi, 2002], [Xiong and Matthies, 1997]. The authors also provide a mathematical analysis of the sub-pixel locking effect for SSD and SAD similarity measures in combination of quadratic curve fitting and show that this effect is increased if the imagery contains high frequency information. [Shimizu and Okutomi, 2001] provide an algorithm for easing sub-pixel locking which requires a recalculation of the DSI based on re-interpolation of match images at half pixels positions. Disparity estimations are then combined with disparity estimations from the original DSI to cancel pixel locking errors. Unfortunately this approach doubles processing demands. Another technique to improve sub-pixel disparities is up-sampling the imagery to be matched. No need to say that this strategy comes at the cost of significantly increased computational and memory demands. [Stein et al., 2006] reduce sub-pixel locking by avoiding quadratic curve fitting. Inspired by [Lucas and Kanade, 1981], [Tomasi and Kanade, 1991] they utilize integer disparity values as an the initial guess which are further adjusted by minimizing the quadratic error of intensity differences (SSD). At the same time, foreshortening effects are compensated by adjusting windows shapes by affine warping to a planar surface approximation. Note that there is a close relation to the adaptive least squares correlation method proposed by [Gruen, 1985]. Due to the fact that all these methods involve considerable computational effort we stick to a simple parabola fit. Moreover in most applications we are not restricted to plain stereo and exploit redundancy across multiple images for disparity refinement.

2.2.3 Filter Techniques in Dense Stereo

To identify spurious correspondences numerous post-processing steps were developed. A standard technique to remove false matches is simple median filtering for which efficient implementations are available. A challenging topic which has drawn much attention is the detection of occluded areas. This concerns object surfaces which are seen in one view but occluded in the other. In the view in which the area is observed dense matchers tend to over-smooth due to smoothness assumptions in the optimization step which leads to mismatches. For an excellent overview of approaches easing these problems and respective evaluations see [Egnal and Wildes, 2002]. Probably the most common filter is the left-right check (LRC) based on the assumption that resultant disparities of matching first to second image are consistent with theses of matching second to first image: $\|D'(x + D(x, y), y) + D(x, y)\| < 1$. Other algorithms are based on Match Goodness Jumps (MGJ). This is basically the assumption that responses of similarity measures are low in regions of occlusions. Theses areas are identified and disparities invalidated. Another filter technique known as the ordering constraint (ORD) assures that if a point $\mathbf{x}$ is left of the point $\mathbf{y}$ in the left image then $\mathbf{x}'$ is left of $\mathbf{y}'$ in the second image. Disparity jumps in the first view induce a certain number of pixels which are skipped in the second view. This is modeled by the occlusion constraint (OCC) which identifies and invalidates skipped pixels.

2.3 Consistent Surface Models From Point Clouds and Depth Maps

Whereas many algorithms in MVS and DSM generation directly produce consistent surfaces, depth map based methods require an additional fusion step which merges the sub-reconstructions (depth maps) into one consistent model. Since imagery for the purpose of image-based 3D reconstruction typically is collected using large overlaps to guarantee good quality of matching and to avoid data holes, the resultant depth maps do overlap as well. In the course of depth map fusion one aims at improving precision of reconstructed surfaces, data reduction and perhaps most important removing outliers. Latter reduces manual user interaction significantly. Fusion of depth maps and point clouds has been an active research topic for decades primarily in the computer vision and the graphics community. In this section we review related work for algorithms producing true 3D surfaces, however algorithms for image-based 2.5D reconstruction were discussed in section 2.1.6.

A large portion of depth map fusion algorithms builds up on volumetric range integration of depth maps (VRIP) [Curless and Levoy, 1996]. Typically a signed distance field is computed on a (multi-level) octree structure by projection of depth estimations from which then a triangulation can be derived for example using the Marching Cube algorithm [Lorensen and Cline, 1987]. A recent voxel based approach was proposed in [Zach et al., 2007] using depth maps to construct a truncated signed distance field. The surface represented by a level set is extracted by minimization of a TV-L1 based global energy. Thereby the total variation of the level set, which is a measure for the surface perimeter, and a data term which represents the absolute variation of the level set and the signed distance field is minimized. Using the L1-norm leads to increased robustness in presence of outliers. Despite impressive results and the possibility of parallel execution on GPUs ([Zach, 2008]) the time and memory demands are significant. Moreover, depth samples across views possessing different scales is challenging for VRIP approaches since operating on constant voxel sizes. One example addressing this issue is the scale space representation proposed in [Fuhrmann and Goesele, 2011]. They build a multi-level octree holding vertices at different scales. Depth observations from the depth maps are inserted according to their pixel footprint. This way a hierarchical signed distance field is generated. Regularization is applied by interpolating depths from more confident samples from coarser levels. For iso-surface extraction the most detailed surface representation is preferred.

A computational effective approach is the ball pivoting algorithm proposed by [Bernardini et al., 1999]. Starting with a seed triangle a sphere of a user-defined size is pivoted around each edge, until another point is touched. This point and the vertices of the edge define a new triangle. This procedure is repeated until no more points can be assigned and the a new seed triangle is selected. The algorithm terminates as soon as all points have been assigned.

An algorithm producing watertight meshes of excellent quality was proposed by [Kazhdan et al., 2006], [Kazhdan and Hoppe, 2013]. The algorithm operates on oriented point sets and models the surface as an indicator function evaluating to 1 behind the surface and 0 in front of the surface. At the in-front / behind transition the gradient of the vector field is maximal and in areas not part of the surface the gradient evaluates to 0. Oriented points can be considered as samples of the indicator functions gradient and are used to construct a gradient vector field $\mathbf{V}$. The indicator function $\mathcal{X}$ is given by the function minimizing the absolute difference between $\mathbf{V}$ and the indicator functions gradient $\nabla\mathcal{X}$, thus $min_{\mathcal{X}}\|\nabla\mathcal{X} - \mathbf{V}\|$. This problem is further transformed by applying divergence operators of both, $\mathbf{V}$ and $\nabla\mathcal{X}$ which translates the minimization into a Poisson problem. In practice the problem is discretized using a multi-level octree to represent the vectorfield $\mathbf{V}$. For each leaf node the $\mathbf{V}$ is computed based on the sample positions and normals taking into account the distance to node centres and the node level. The solution to the Poisson problem can be computed by solving a sparse linear system possessing the dimensions equal to the number of nodes.

[Labatut et al., 2009] cast the problem of depth map fusion in a energy minimization problem solved by a s-t cuts optimization framework. Thereby points generated from the depth maps are subject to a Delauney triangulation. The dual sub-divides the space into tetrahedra which are then labeled as inside or outside defining the faces representing the surface. This binary labeling is carried out using s-t cuts minimizing lines of sight intersecting the surface and quality of the surface estimated using the concept of β-skeletons [Amenta et al., 1998].

Another type of algorithms utilize the spatial information already contained in the depth maps by triangulation in image space, lifting the results to object space and stitching single meshes. A purely geometric algorithm for depth map merging is Polygon Zippering proposed in [Turk and Levoy, 1994]. The method generates triangle meshes by simply constructing two faces from four adjacent depth estimations. Suspicious triangles are removed by evaluation of the triangle side lengths. After alignment of meshes, redundant triangles are removed from the boundaries of single patches and remainders are connected. Redundancy can be exploited by mean or median operations, however, visibility constraints are not enforced. [Merrell et al., 2007] proposed a method for the fusion of noisy depth maps in real-time applications. Proximate depths maps are rendered into one reference view. Redundant depths per pixel are checked for geometric consistency and are filtered using occlusion and confidence checks. After consistent depth estimations are averaged, a mesh is constructed on the depth maps using quadtrees and lifted to 3D space.

Chapter 3

Overview of the Reconstruction Process

Before we explain our algorithms in detail we give a rough overview of the implemented reconstruction pipeline. We presume a set of input images for which interior and exterior orientations are known. The algorithmic modules are displayed in figure 3.1. Within a first step lens distortions are removed from the images. Typically information of lens distortions are estimated within a calibration procedure using laboratory test fields or are determined within a self calibration process in the course of BA. The implemented pipeline covers undistortion capabilities for several distortion models foremost the Brown model[Duane, 1971] which often is utilized by commercial and academic software packages for SfM and BA. The set of undistorted images $\mathbf{I}_n$ $n = 1, ..., N$ along with their orientations serve as input of the depth map based reconstruction process. Our depth map generation involves dense stereo matching which presumes the selection of suitable image pairs. On the one hand it does not make sense to match each image against all other images because in general different parts of the scene are mapped. Even if identical scene parts are mapped, viewing direction of two cameras and therefore image content might considerably vary such that matching would fail due to insufficient image similarity. On the other hand if baselines are to small image matching works well but the geometric properties for forward intersection is poor, which results in noisy surface points. Whereas latter problem is tackled during the forward intersection algorithm itself, the identification of suitable stereo pairs providing a certain degree of image similarity is implemented within the stereo selection module which will be explained in section 3.1. Once suitable stereo pairs are selected, depth maps are generated incorporating dense stereo matching, subsequent correspondence linking and multi-baseline triangulation. In the following, we use the naming as defined in section 2.2 of base/reference/master image and match/search/slave image synchronously. Within our implementation each image $\mathbf{I}_n$ of the block is treated as base image $\mathbf{I}_b$ and matched against a set of match images $\mathbf{I}_{m,i}$ as identified by the model selection process. Subsequent to stereo matching, redundant depth estimates of the single stereo models $(\mathbf{I}_b, \mathbf{I}_{m,i})$ are linked base image wise and utilized for multi-view forward intersection. This way for each reference view $\mathbf{I}_n$ a depth map $\mathbf{D}_n$ is generated. We refer to this process as depth map generation. The general workflow will be outlined in section 3.2, an in depth discussion of all involved algorithms is given in chapter 4. Since spatially neighboring base images generally cover the same scene extent, the resulting depth maps or point clouds still hold redundant information of the surface. In order to get a consistent representation of the scene they are fused in the final processing step leading to either 2.5D elevation maps or 3D point clouds or triangle meshes. Again, in section 3.3 a general overview is given and the two strategies are discussed in depth in chapters 5 and 6. A flow chart of the general processing pipeline is depicted in figure 3.1.

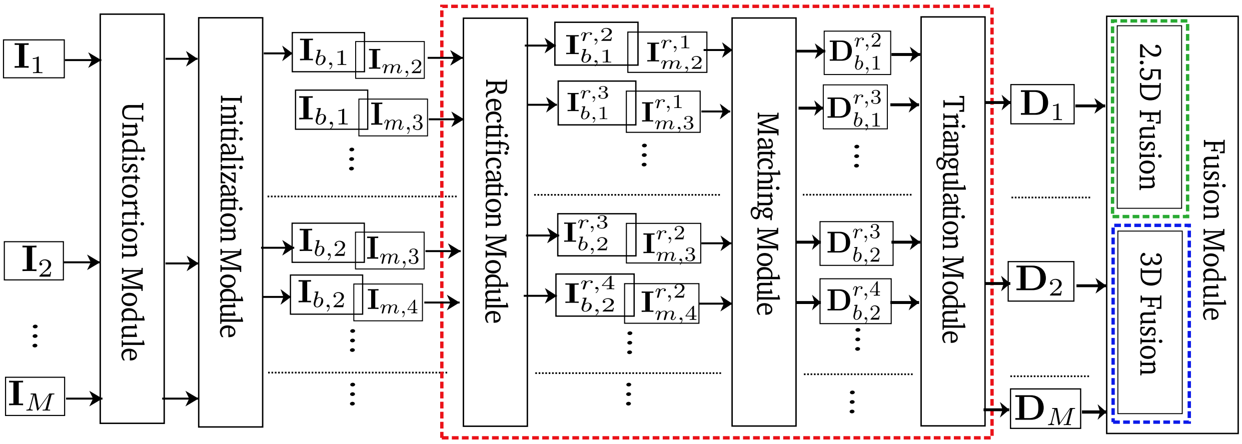

$\mathbf{I}_1$
$\mathbf{I}_2$
$\mathbf{I}_M$
Undistortion Module
Initialization Module
$\mathbf{I}_{b,1}$ $\mathbf{I}_{m,2}$
$\mathbf{I}_{b,1}$ $\mathbf{I}_{m,3}$
$\mathbf{I}_{b,2}$ $\mathbf{I}_{m,3}$
$\mathbf{I}_{b,2}$ $\mathbf{I}_{m,4}$
Rectification Module
$\mathbf{I}^{r,2}_{b,1}$ $\mathbf{I}^{r,1}_{m,2}$
$\mathbf{I}^{r,3}_{b,1}$ $\mathbf{I}^{r,1}_{m,3}$
$\mathbf{I}^{r,3}_{b,2}$ $\mathbf{I}^{r,2}_{m,3}$
$\mathbf{I}^{r,4}_{b,2}$ $\mathbf{I}^{r,2}_{m,4}$
Matching Module
$\mathbf{D}^{r,2}_{b,1}$
$\mathbf{D}^{r,3}_{b,1}$
$\mathbf{D}^{r,3}_{b,2}$
$\mathbf{D}^{r,4}_{b,2}$
Triangulation Module
$\mathbf{D}_1$
$\mathbf{D}_2$
$\mathbf{D}_M$
2.5D Fusion
3D Fusion
Fusion Module

Figure 3.1: Flow chart of the implemented pipeline. After undistortion within an initialization module suitable stereo pairs to be incorporated into the reconstruction process are selected. Selected stereo pairs then are rectified matched. Eventually disparity maps are triangulated base image wise resulting in one depth map $\mathbf{D}_j$ per image. In a last step depth maps are fused. Dependent of the desired output 2.5D elevation maps or 3D triangle meshes are generated. Algorithms marked by the red box will be discussed in chapter 4. 2.5D fusion (green box) and 3D fusion (blue box) will be discussed in chapters 5 and 6 respectively.

3.1 Model Selection

The quality of stereo matching is highly dependent on the radiometric similarity of the two incorporated images. As mentioned before images within the same block generally map different parts of the scene, or even if the same scene extent is mapped, viewing directions might vary considerably. As a result image similarity is low, a large number of half occlusions occur and the matching process fails or delivers poor results. Within the model selection process suitable stereo pairs are selected which then form the input for the main processing pipeline. We base the selection criterion on the density of reconstructed pixels. According to [Seitz et al., 2006] we follow a quasi-geometric approach in terms of occlusion handling. In order to reduce time, the complete selection procedure is carried out on low resolution imagery. In contrast to other approaches which directly utilize sparse feature points available from the SfM/BA procedure (for example [Furukawa and Ponce, 2010], [Goesele et al., 2007], [Hiep et al., 2009]) to establish the connectivity between multiple frames we unfortunately can not presume this information. Therefore we analyze the connectivity across frames by dense matching low-resolution versions of the available imagery. Let $\mathbf{I}_b$ a base/reference/master image from the set of all images $\mathcal{N}$, we wish to identify the set $\mathcal{N}_b \subseteq \mathcal{N}$ of suitable match/search/slave images $\mathbf{I}_{m,i} \in \mathcal{N}_b$. Within a first step relative translations are analyzed. Therefore the baselines of all view tuples $(\mathbf{I}_b, \mathbf{I}_{m,i})$, $\mathbf{I}_{m,i} \in \mathcal{N}$ are computed and the n_b tuples possessing the shortest baselines form the first selection of match images denoted by the set $\mathcal{N}_f$. Then the selected tuples $(\mathbf{I}_b, \mathbf{I}_{m,i})$, $\mathbf{I}_{m,i} \in \mathcal{N}_f$ are matched using the SGM based stereo approach explained in detail in section 4.2. Due to the low resolution of imagery the matching process is very fast. Since we are only interested in the common scene coverage, low precision and lack of detail can be accepted during this step. As the measure of mutual scene coverage we employ the number of successfully matched pixels. More precisely, if the ratio of successful matched pixel to all image pixels is above a threshold t_p the tuple is considered to be a suitable stereo pair and $\mathbf{I}_{m,i} \in \mathcal{N}_b$, else the tuple is discarded from further processing. Typically we choose rather conservative thresholds $n_b = 20$ and $t_p = 0.2$ for unstructured data sets. This assures that the a relatively large number of stereo pairs are selected and pairs potentially contributing to the surface reconstruction are not discarded. However, if flight patterns are known, as most often when processing airborne nadir data sets, the thresholds are adapted according to the the specific pattern. This adaption is based on pre-knowledge considering the trade off between time requirements, precision improvement and reconstruction density.

3.2 Depth Map Computation

Based on the tuples $(\mathbf{I}_b, \mathbf{I}_{m,i})$, $\mathbf{I}_{m,i} \in \mathcal{N}_b$ as available from the initialization process for each reference image $\mathbf{I}_b$ a depth map or point cloud is generated. This process is depicted by the red box in figure 3.1 and involves pair-wise rectification and matching of $(\mathbf{I}_b, \mathbf{I}_{m,i})$. The rectification process can be considered as a preprocessing step prior to the actual depth reconstruction. The basic idea is to utilize restrictions from epipolar geometry to re-sample a pair of images such that epipolar lines are horizontal and homologous image coordinates are located in identical rows. On the one hand this simplifies algorithms for dense stereo and on the other hand speeds up the matching process. Different types of rectification will be explained in detail in section 4.1. We put emphasis on the topic because the proposed forward intersection methods depend on the type of utilized rectification method. For the further discussion we enhance previous notation of base and match images as follows. Let $\mathbf{I}_{b,j} \in \mathcal{N}$ be a base image, where the base image role is denoted by the first element of the lower index, and $j = 1..M$ identifies the image in $\mathcal{N}$. Similarly, let $\mathbf{I}_{m,i}$ be a match image, with $\mathbf{I}_{m,i} \in \mathcal{N}_{b,j}$ with $i = 1..N$ identifying the image index in the subset $\mathcal{N}_{b,j}$ of reference view dependent match images. Pairwise rectification of a reference view j leads to N rectified image tuples $(\mathbf{I}^{r,i}_{b,j}, \mathbf{I}^{r,j}_{m,i})$, see figure 3.1. The first upper index r denotes that the image is rectified and the second index denotes the index of the image it is rectified with. The rectified image tuples are subject to SGM-based dense stereo matching as explained in section 4.2. Thereby for each pixel in the base image a disparity is estimated encoding the 3D coordinates of the respective pixel. Since each base image is matched against N images 3D information of surface points visible in the reference view are computed redundantly. This redundancy is utilized for outlier rejection and improvement of precision in the course of forward intersection. Using results from rectification and information from the disparity maps a pixel in the base image $\mathbf{I}_{b,j}$ can be linked to image coordinates of the rectified match images $\mathbf{I}^{r,j}_{m,i}$ as will be shown in section 4.3. This leads to a set of redundant observations for which geometric consistency is verified and multi-view forward intersection is carried out. This multi-baseline triangulation process leads to a depth map $\mathbf{D}_j$ for each base image j. We found that the check for geometric consistency delivers depth maps possessing very small number of outliers which is beneficial for the subsequent fusion steps.

3.3 Depth Map Fusion

In the course of depth map computation for each image $\mathbf{I}_j$ of the data set one depth map $\mathbf{D}_j$ is derived. As the images $\mathbf{I}_j$ overlap, the depth maps $\mathbf{D}_j$ overlap as well. To derive a consistent non-redundant scene representation an algorithm for the fusion of depth maps has to be provided. Redundant observations across depth images in general posses different precisions due to variances in image scale, number of redundant observations incorporated in the multi-baseline triangulation, accuracy of stereo matching itself etc. Paying respect to such precision variances during the fusion process is mandatory to further exploit redundancy for outlier filtering and improvement of surface accuracy. Moreover, algorithms should be designed in a scalable way enabling surface reconstruction of large scale data sets. As can be seen in figure 3.1 within the presented reconstruction pipeline two different fusion modules are implemented. The first approach generates 2.5D elevation maps $f(\mathbf{x}) = f(x, y)$. In other words the scene is represented as a grid with each grid cell providing one height value. For airborne nadir data sets this is a common representation since reconstructed surfaces in $\mathbf{D}_j$ contain only limited 3D structure due to restricted viewing directions. Therefore reconstructed geometry can be represented sufficiently good by 2.5D elevation maps $f(\mathbf{x})$. These maps store spatial topology by design which enables convenient and fast data access for further processing steps as for example classification tasks, feature collection, ortho-photo or DTM generation. However, as camera system possessing wider angles are utilized viewing angles increase and more 3D structure (as for example facade points) is reconstructed. Obviously this information can not be represented by elevation maps which can only hold one height per grid cell. Generally one is interested in the highest elevation per cell and robust filters have to be provided to completely discard lower points whilst performing robust filtering on higher points. Our approach is based on orthographic projection of points onto a plane parallel to the earth surface and subsequent filtering procedures as explained in chapter 5. The second fusion algorithm is designed for 3D surface extraction of image collections not representable by elevation maps. These scenarios often occur for UAV missions with the goal of 3D mapping. The final output is a surface represented by a triangle mesh. Within the 2.5D approach spatial neighborhood of scene points is encoded by a two dimensional surface grid and local surface normals can be approximated by the normal of the grid. For the 3D fusion process such initial normal assumptions are not feasible. Therefore we extract point-wise normals in the depth maps using an geometry adaptive triangulation. Furthermore, more complex 3D structures capturing scale variances for filtering tasks has to be provided. The presented method is based on a multi-scale octree to which all observations are inserted, favouring points at high sampling rates and precision. These points are then median filtered along the surface normal leading to a set of improved oriented points which are suitable for mesh generation.

Chapter 4

Generation of Depth Maps

One of the core elements of the proposed multi-view stereo system is the generation of depth maps. As discussed in chapter 3 the implemented algorithm is based on stereo matching a reference image against a set of overlapping match images $\mathbf{I}_{m,i}$ with $i = 1, ..., N$. This leads to N disparity maps holding depth information for virtually every pixel in $\mathbf{I}_b$. In a final step for each pixel in the base image redundant observations from the single stereo disparity maps are linked and 3D coordinates are derived from multi-baseline forward intersection leading to one depth map D per base image $\mathbf{I}_b$. In this chapter the single algorithms involved in this process are explained in detail. All methods discussed in this chapter are depicted by the red box in figure 3.1.

For DSI-based methods the generation of stereo disparity maps is typically split into two processing steps: rectification of image pairs and subsequently solving the dense correspondence problem for rectified pairs. We assume that the relative orientation between the two views are known and images are distortion-free. Let $\mathbf{x}_b$ be a pixel in the first (master, base) image $\mathbf{I}_b$ and $\mathbf{x}_m$ its correspondence in the second (slave, match) view $\mathbf{I}_m$. Utilizing epipolar geometry the search for $\mathbf{x}_m$ can be geometrically restricted to the epipolar line $\mathbf{l}_m$ in the second view. The correspondence search would involve computation of $\mathbf{l}_m$ for each pixel in $\mathbf{I}_b$, re-sampling along the 2d line $\mathbf{l}_m$ and interpolation of intensities at sampled sub-pixel locations to derive a similarity measure and identify the most probable match. To speed up computation this problem can be transformed to scanning 1D lines instead by rectification of image pairs. Image re-sampling is carried out image-wise such that in the course of correspondence search interpolation of intensities can be avoided. Moreover, the transformation assures that homologous points across two views share the same row index. This transformation process is referred to as image rectification and three different methods are discussed in the section 4.1. We put emphasis on the rectification types since proposed methods for structure computation will depend on the type of rectification and they posses different characteristics with respect to precision and relative orientations of stereo pairs. Once image pairs are rectified the actual correspondence search is carried out. Let $\mathbf{x'}_b$ be a pixel in the rectified base image $\mathbf{I'}_b$ and $\mathbf{x'}_m$ be a pixel in the rectified base image $\mathbf{I'}_m$. Since rectification assures identical row indexes of homologous image coordinates $\mathbf{x'}_b(x'_b, y')$, $\mathbf{x'}_m(x'_m, y')$ a correspondence can be conveniently encoded by the disparity or parallax d such that the link of homologous entities is encoded by $\mathbf{x'}_b(x', y')$, $\mathbf{x'}_m(x' + d, y')$. Thus the pixel-wise 3D information of an image can be represented by the disparity image $D(\mathbf{x})$, holding disparities of each pixel. The aim of dense stereo matching is to estimate the most probable pair of pixel correspondences for each pixel across two views. For each pixel the most probable candidate from a set of geometrically possible candidates has to be identified based on image similarity and some sort of smoothness assumption. For this task we apply a modification of the SGM algorithm limiting disparity and image search spaces by a coarse-to-fine processing scheme. For each base image, the algorithm outputs pixel-wise disparity maps $D_{b,i}$ as will be explained in detail in section 4.2. The set of N base image-wise disparity maps are subject to correspondence linking, outlier rejection based on geometric consistency and multi-baseline forward intersection. In the course of correspondence linking redundant depth estimates are collected from the respective set of depth maps. The main goal is exploiting the available redundancy to improve the quality of reconstructed surfaces. Details of the triangulation approach will be explained in section 4.3.

4.1 Rectification of Calibrated Image Pairs

Image rectification is the process of transforming a pair of images $\mathbf{I}_b$ and $\mathbf{I}_m$ in a way that all pairs of homologous image points $\mathbf{x}_b$ and $\mathbf{x}_m$ are remapped to the identical row or column in the rectified images $\mathbf{I}'_b$ and $\mathbf{I}'_m$. Without loss of generality we restrict the following discussion on mapping homologous image coordinates to identical rows. Therefore we apply a remapping function Φ such that

$$\begin{aligned} \phi(\mathbf{x}_b(x_b, y_b)) &= \mathbf{x}'_b(x'_b, y') \\ \phi(\mathbf{x}_m(x_m, y_m)) &= \mathbf{x}'_m(x'_m, y'). \end{aligned} \tag{4.1}$$

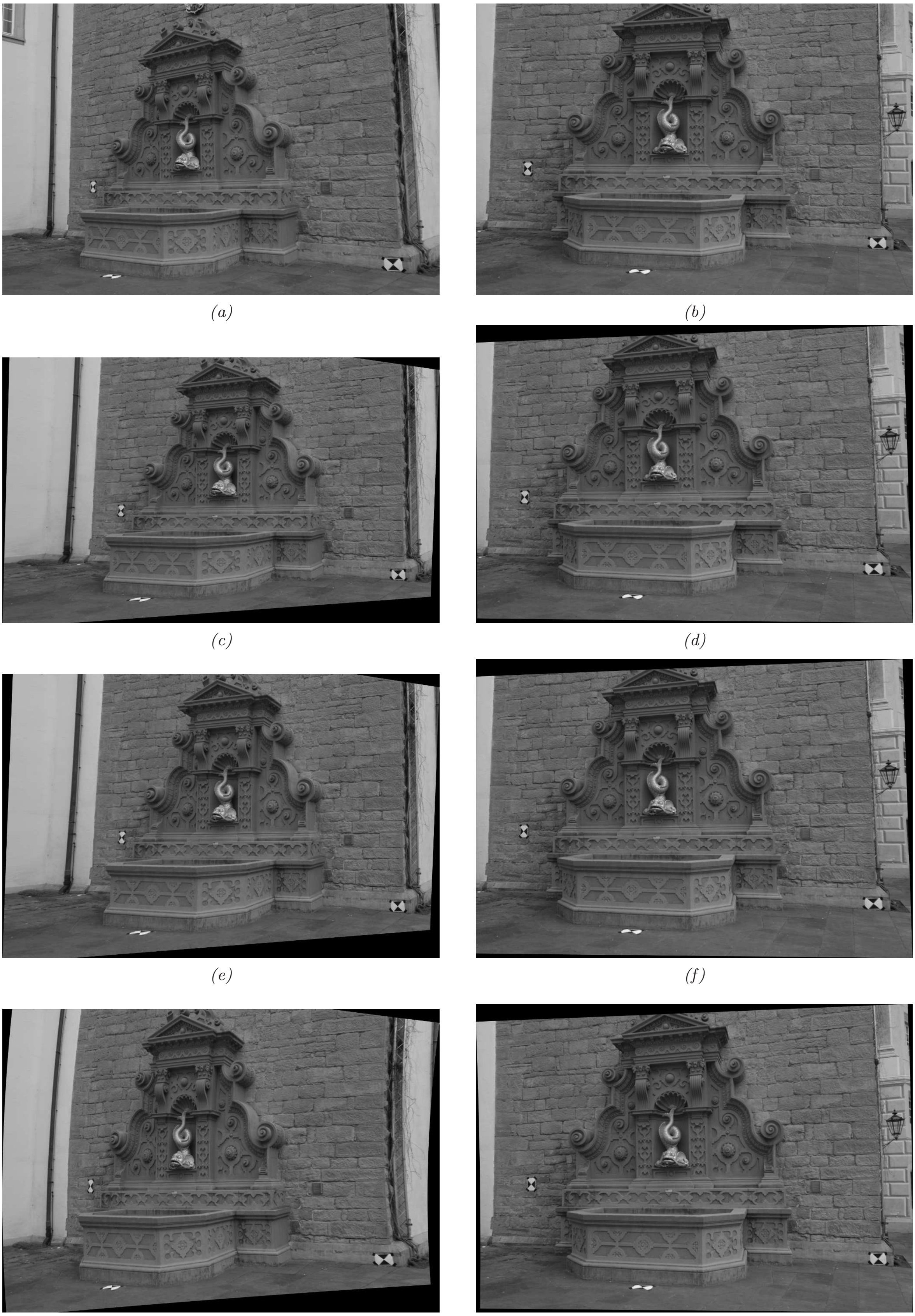
Beside pairwise rectification methods, algorithms for triples of images exist [Ayache and Hansen, 1988] but, due to minor practicability for our multi-view algorithm, will not be discussed in this work. Independent of the rectification method, computation of the remapping function Φ is based on epipolar geometry, thus accuracy of rectified images is dependent of the precision of relative orientation. Within this thesis three different approaches for image rectification were implemented. We distinguish two types of rectification strategies. The algorithms proposed in [Fusiello et al., 2000] and [Loop and Zhang, 1999] are based on remapping a pair of images on virtual image planes utilizing homographies. In contrast to the first approach the second method seeks to define virtual image planes minimizing the resulting projective distortions. However, homography-based algorithms suffer from the limitation of not being capable to rectify image pairs of arbitrary camera movement. More precisely, they fail for stereo configurations close to pure forward motion, that is motion in viewing direction. This limitation was tackled by the algorithm proposed in [Pollefeys et al., 1999] introducing the concept of half-epipolar spaces. In section 8.2 we discuss the algorithms in more detail. Emphasis is put on the rectification process since we build up on results within the proposed triangulation methods. An evaluation of the impact on dense stereo matching is given in the result section 4.4.2.

4.1.1 Examples

In this section rectification results of two example stereo pairs are given. For a quantitative evaluation of the actual influence of rectification on dense matching results the reader is referred to the evaluation section 4.4.2. The first example is a typical side-ward motion stereo pair as depicted in figure 4.17a and 4.17b. Image dimensions of all rectified pairs are rather similar. However, for polar rectified pairs slightly more distortion can be observed. Figures 4.2a and 4.2b depict an example stereo pair of the forward motion configuration. Whereas homography-based rectification was not capable to produce rectified imagery decent results can be obtained employing polar rectification. Fussielo's and Loop's rectification would produce images possessing huge image dimensions (13681 × 14467 pixels) and/or large distortions hindering dense stereo matching. Evaluations in chapter 4.4.2 show that slightly more precise results at higher densities are obtained by using homography based approaches. Note that this might only partly be due to the rectification methods but also due to the different approaches of forward intersection. Therefore we prefer to utilize homography based approaches but use polar rectification as fall-back solution for in-motion configurations.

4.2 SGM-based Dense Matching

In this section the implemented algorithms for dense image matching are discussed. We assume input imagery to be rectified by one of the methods described in sections 8.2.1, 8.2.2. The implemented stereo method is based on SGM but extends the classic approach as proposed in [Hirschmüller, 2008] by pixel-wise adapted disparity search ranges, estimation of commonly seen image areas, improved optimization schemes and an automatic tiling strategy adapting to hardware resources and minimizing processing overhead. The



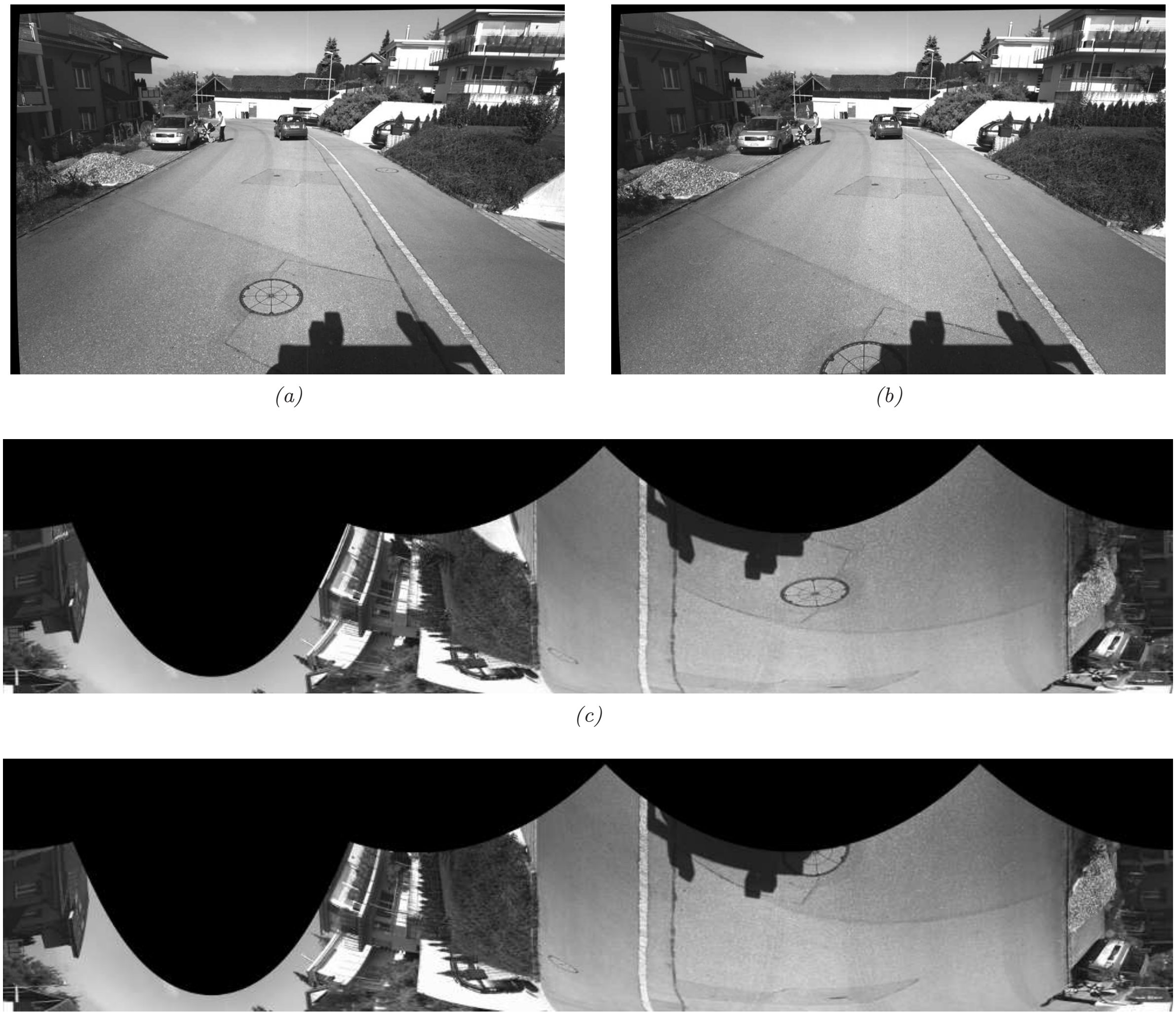
(a) *(b)*

(c) *(d)*

(e) *(f)*

(g) *(h)*

Figure 4.1: Results of the three rectification methods for an image pair of the Fountain data set [Strecha et al., 2008]. (a),(b): original images. (c),(d) homographie based [Fusiello et al., 2000], (e),(f) homographie based [Loop and Zhang, 1999]. (g),(h) polar rectification



(a) *(b)*

(c)

(d)

Figure 4.2: Results of polar rectified imagery for forward-motion configuration. data set [Strecha et al., 2008]. (a),(b): original images. (c),(d) polar rectified images (vertical epipolar lines)

key advantages are reduced computational complexity, reduced memory consumption and the ability of processing scenes without previous knowledge about depth or disparity ranges. Furthermore, ambiguities of photo consistency measures due to weak or high frequent texture are resolved more reliably.

Review of the SGM algorithm

The problem of dense stereo matching can be stated as finding pixel-wise correspondences $\mathbf{x}_b = [x_b, y_b]$, $\mathbf{x}_m = [x_m, y_m]$ across two views representing the same world object for all pixels in the images. Operating on rectified images, potential correspondences (representing the same world object) are located in the same row of the base or master image $\mathbf{I}_b$ and the match or slave image $\mathbf{I}_m$, in other words $y_b = y_m$. Therefore the problem can be reformulated as finding the disparity $d = x_m - x_b$. In order to select the most probable correspondence from the set of potential correspondences photo-consistency measures are utilized. These measures define a cost $C(d)$ for each pair of pixels and possess low values if the pixel and its neighbourhood are similar and increases as intensity variations in the respective image regions increase. However, due to perspective distortions, illumination changes, non-Lambertian surfaces, lack of texture, etc. these costs can not be assumed to give reliable responses in general. Therefore, in most of the state-of-the-art stereo algorithms local costs based on photo consistency measures are enhanced by shape priors, forcing neighbouring disparities, and therefore the extracted surface, to be smooth. The problem of deriving disparities minimizing photo consistency costs and simultaneously forcing shape priors for all image pixels typically is formulated as an global energy minimization problem of the form:

$$E = E_{data} + E_{prior}. \tag{4.2}$$

More specifically for dense stereo problems the energy functional can be stated as

$$E(\mathbf{D}) = \sum_{\mathbf{x}} E_{data}(d_{\mathbf{x}}) + \sum_{\mathbf{y} \in \mathcal{N}_{\mathbf{x}}} E_{prior}(d_{\mathbf{x}}, d_{\mathbf{y}}) \tag{4.3}$$

where E_{data} is based on the costs induced by photo consistency and E_{prior} penalizes large variations of $d_{\mathbf{x}}$ and its surrounding disparities $d_{\mathbf{y}}$. Unfortunately the problem is considered to be NP hard [Boykov et al., 2001], therefore is not solvable in polynomial time. However, various algorithms have been proposed to compute approximations of the global minimizer for these sorts of energies, most of them based on loopy belief propagation or graph cut based techniques (see section 2.1.5). The solutions are not guaranteed to be optimal but approximate the global optimum sufficiently good. Unfortunately, most energy minimization techniques are computational expensive and require large amount of memory. This is a problem in particular for reconstruction using large frame aerial imagery. The SGM algorithm proposed in [Hirschmüller, 2008] favorably compares on hardware requirements and computational complexity. Its basic idea is to apply dynamic programming / scanline optimization along multiple directions of the image. The result defines a set of disparities minimizing a global cost function of the form

$$\begin{aligned} E(D) = &\sum_{\mathbf{x}_b} (C(\mathbf{x}_b, D(\mathbf{x}_b)) \\ &+ \sum_{\mathbf{x}_{\mathcal{N}}} P_1 T[\|D(\mathbf{x}_b) - D(\mathbf{x}_{\mathcal{N}})\| = 1]_. \\ &+ \sum_{\mathbf{x}_{\mathcal{N}}} P_2 T[\|D(\mathbf{x}_b) - D(\mathbf{x}_{\mathcal{N}})\| > 1] \end{aligned} \tag{4.4}$$

Thereby $D(x, y)$ represents the disparity image holding disparity estimates of all base image pixels $\mathbf{x}_b$. The first term is the data term composed of the pixel-wise photo consistency costs C computed from the potential correspondences $(\mathbf{x}_b, d)$. The latter two terms represent the smoothness term. This type of formulation is known as the linear truncated penalty term [Boykov et al., 1998]. T is an operator evaluating to one if the subsequent condition is true and evaluates to zero else. $\mathbf{x}_N$ denote base image pixels in the neighbourhood

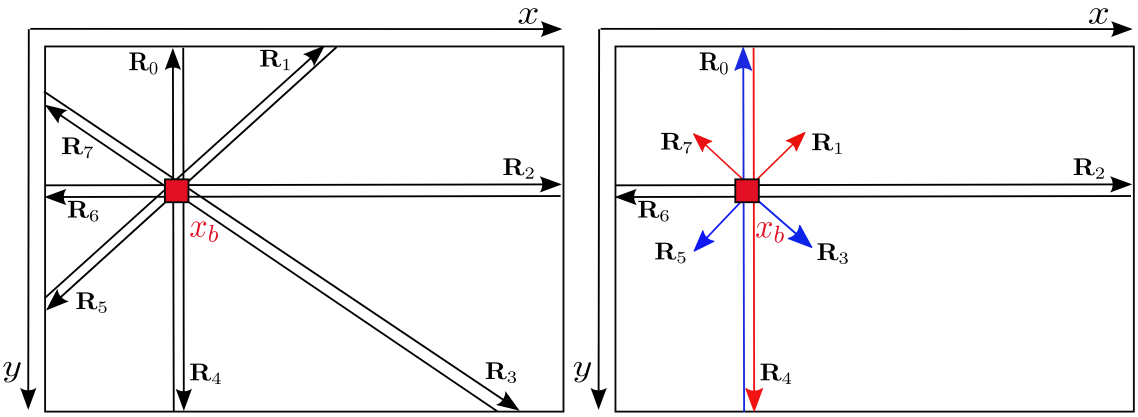


Figure 4.3: Left: Visualization of the 8 image paths along which costs for the pixel $\mathbf{x}_b$ are accumulated. Right: Visualization of the accelerated cost aggregation. Instead of passing the DSI eight times along the paths R_{0-7} the DSI is passed only 4 times along R_0,R_2,R_4,R_6. When passing along R_0 the costs from the directions R_3 and R_5 are accumulated at the same time. Similar procedure is applied for path R_4.

of $\mathbf{x}_b$. The penalty parameters P_1 and P_2 control the gain of surface smoothing. Thereby P_2 is assumed to be larger as P_1. Computation of the disparity image D approximating the solution minimizing the global energy 4.4 is carried out in two steps. First a DSI (Disparity Space Image) is generated holding the local costs $C(\mathbf{x}_b, d)$ for each base image pixel and its set of potential correspondences. Thereby d is a discrete value in a constant range $d \in [d_{min}, d_{max}]$ defining all the potential correspondences along the epipolar line. Each $C(\mathbf{x}_b, d)$ is assigned to the DSI, a three dimensional cube structure of the dimensions $r \times c \times (d_{max} - d_{min} + 1)$. In the following we denote this structure by $C(\mathbf{x}, d)$. As mentioned before, selecting disparities based on photo consistency costs solely would yield wrong results for surface parts possessing challenging texture. Therefore, in the second step costs are accumulated along i (typically $i = 8$ or $i = 16$) image paths. The resulting accumulated costs are stored in a 3D structure $S(\mathbf{x}, d)$ with the same dimensions as $C(\mathbf{x}, d)$. The accumulation process is similar to dynamic programming but instead of accumulating costs only in one direction (traditionally along epipolar lines), accumulation is carried out sequentially in multiple directions as visualized in figure 4.4. However, the ordering constraint, as utilized in dynamic programming approaches, can not be forced along paths differing from the epipolar line. The accumulation procedure along one of the i paths specified by $\mathbf{R}_i$ can be recursively formulated as

$$\begin{aligned} L_{\mathbf{r}_i}(\mathbf{x}_b, d) =& C(\mathbf{x}_b, d) + min(L_r(\mathbf{x}_b - \mathbf{r}_i, d), \\ & L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d-1) + P_1 \\ & L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d+1) + P_1, \\ & L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, i) + P_2) \\ & - min_k L_r(\mathbf{x}_b - \mathbf{r}_i, k). \end{aligned} \tag{4.5}$$

Thereby $\mathbf{r}_i$ is an offset on the path $\mathbf{R}_i$ to the previous pixel. Starting at the image borders $L_{\mathbf{r}}(\mathbf{x}_b, d)$ is initialized with the predefined maximal costs C_{max}. For the next pixel $\mathbf{x}_b$ on $\mathbf{r}_i$ these values, now denoted by $L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d)$ are utilized to compute the current cost string $L_{\mathbf{r}}(\mathbf{x}_b, d)$ for all disparities d. In this manner all $\mathbf{x}_b$ are skimmed on the path until the border of the image is reached. The last term in equation 4.5 ensures that L is upper-bounded by $L_{\mathbf{r}_i}(\mathbf{x}_b, d) < C_{max} + P2$ as the path $\mathbf{R}_i$ is processed. The sum over all paths

$$S(\mathbf{x}_b, d) = \sum_{\mathbf{R}_i} L_(\mathbf{x}_b, d) \tag{4.6}$$

results in a three dimensional structure holding costs for each pixel and its set of potential correspondences. Limiting the upper bounds of $C_{max} + P2$ by by an integer value of 2^{11} the final accumulated costs S are

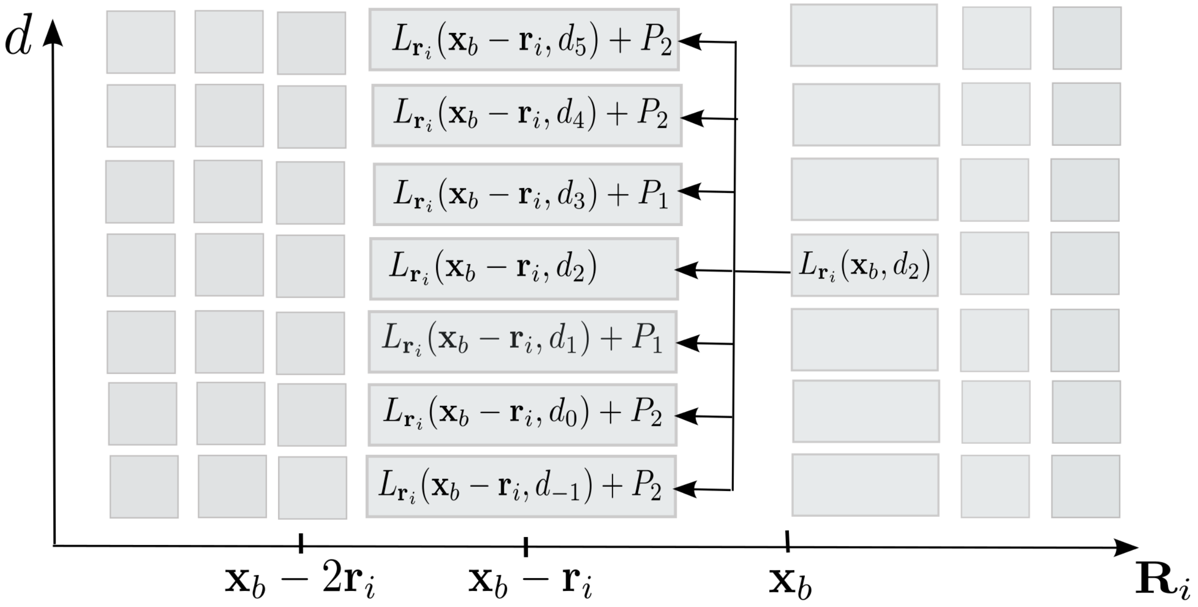

d
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_5) + P_2$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_4) + P_2$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_3) + P_1$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_2)$
$L_{\mathbf{r}_i}(\mathbf{x}_b, d_2)$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_1) + P_1$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_0) + P_2$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_{-1}) + P_2$
$\mathbf{x}_b - 2\mathbf{r}_i$
$\mathbf{x}_b - \mathbf{r}_i$
$\mathbf{x}_b$
$\mathbf{R}_i$

Figure 4.4: Path accumulation for classical SGM on constant disparity ranges. The accumaualted cost $L_{\mathbf{r}_i}(\mathbf{x}_b, d_2)$ is computed using accumulated costs from the pixel $\mathbf{x}_b - \mathbf{r}_i$ on $\mathbf{R}_i$. For all costs in a constant disparity range memory is allocated (depicted by gray boxes) and cost aggregation is performed.

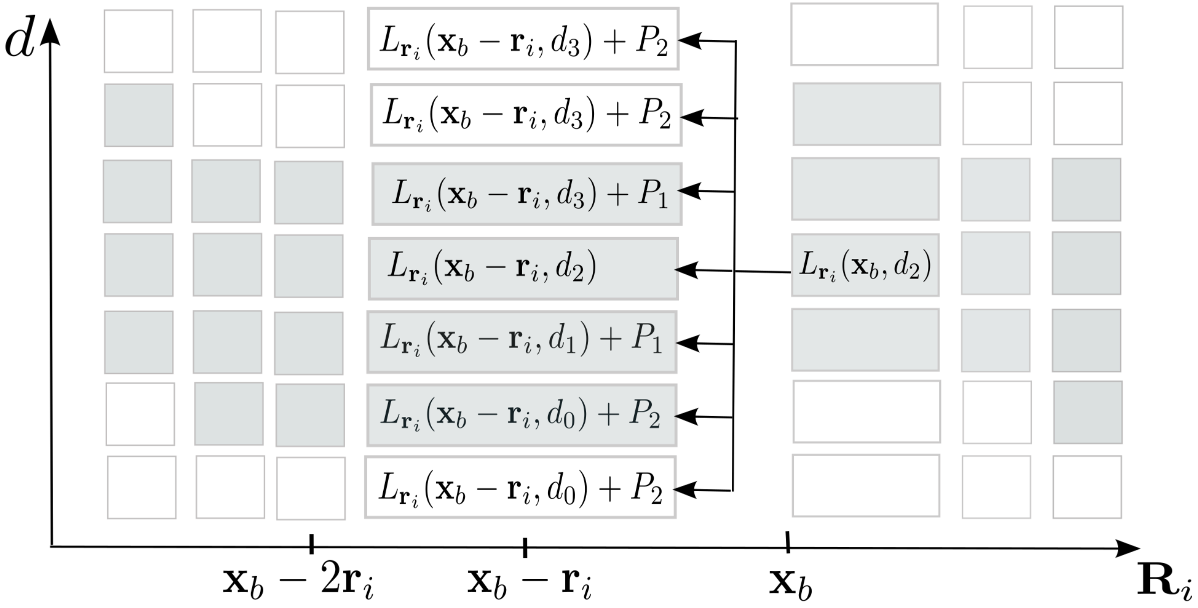

d
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_3) + P_2$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_3) + P_2$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_3) + P_1$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_2)$
$L_{\mathbf{r}_i}(\mathbf{x}_b, d_2)$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_1) + P_1$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_0) + P_2$
$L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_0) + P_2$
$\mathbf{x}_b - 2\mathbf{r}_i$
$\mathbf{x}_b - \mathbf{r}_i$
$\mathbf{x}_b$
$\mathbf{R}_i$

Figure 4.5: Path accumulation for tSGM using dynamic disparity ranges. Costs which are not contained in the dynamic string are approximated by costs from maximal and minimal disparities. Gray boxes mark costs for which memory is allocated and cost aggregation is performed.

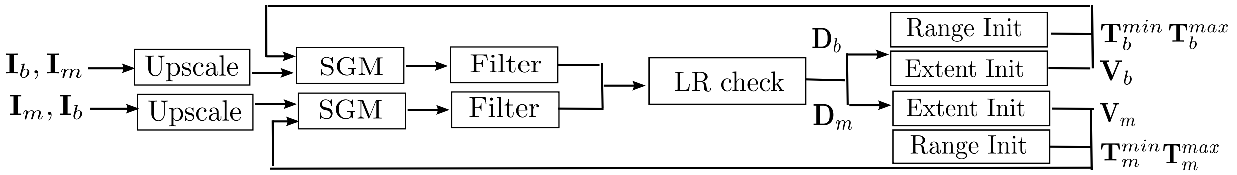


Figure 4.6: Flow chart of our tSGM algorithm. Rectified intensity images $\mathbf{I}_b$ and $\mathbf{I}_m$ are matched and filtered. This process is carried out in parallel with the role of base and match image interchanged. A Left-Right consistensy check leads to the disparity maps D_b and D_m. They serve for the computation of the range images T_b^{min}, T_b^{max}, T_m^{min}, T_b^{max} holding information about minimium and maximum of disparities for initializing search ranges of the next matching iteration. For processing the next lower pyramid intensity images are scaled accordingly.

guaranteed to be smaller than 2^{16}, thus specifying $\mathbf{S}$ as 16 bit array is sufficient. Identifying the minimal aggregated cost $d_{final} = min_d S(\mathbf{x}_b, d)$ for each base image pixel $\mathbf{x}_b$ leads to the final disparity image D approximating the minimizer of the functional (4.4).

Modifications of the SGM algorithm - tSGM

In this section we discuss the implemented methods for stereo matching. An flow chart of the single sub-algorithms is depicted in figure 4.6. Within the original SGM implementation [Hirschmüller, 2008] a coarse-to-fine approach was proposed to initialize and refine entropies and cross-entropies to compute the MI matching cost. Initial disparity images were computed by matching high level (low resolution) image pyramids. The resulting disparities were then used to update the MI matching cost for processing the subsequent pyramid level. In contrast, within our implementation we additionally utilize disparity maps from low resolution matching to restrict disparity search ranges and to identify mutually covered image regions across the stereo pairs. This requires some adaption of algorithms for the computations of photo consistency costs and cost aggregation. Moreover, we discuss the utilized photo consistency measure and implementation details to further speed up processing.

Hirarchical pixel-wise disparity search range restrictions As mentioned before, within the publication [Hirschmüller, 2008] a hierarchical approach was proposed to initialize and update the MI matching cost. We apply the same coarse-to fine strategy, but additionally utilize disparity maps from low resolution matching to restrict disparity search ranges for subsequent matching cycles as visualized in figure 4.6. Let l be the first image pyramid to be processed. After downscaling, the images $\mathbf{I}_b$ and $\mathbf{I}_m$ are subject to SGM stereo matching. This involves the generation of the DSI $C(\mathbf{x}_b, d)$, cost accumulation and selection of the disparities $D(\mathbf{x}_b)$ given by the minimal costs in accumulated cost structure $S(\mathbf{x}_b, d)$. Thereby the disparities search ranges for a pixel $\mathbf{x}_b(x_b, y_b)$ are set to maximal ranges covering the whole image $d_{min}(\mathbf{x}_b) = -x_b$ and $d_{max} = n_c - x_b$ with n_c the number of image columns. This process is carried out, treating $\mathbf{I}_b$ as reference image and image $\mathbf{I}_m$ as match image and in parallel with converse roles of $\mathbf{I}_b$ and $\mathbf{I}_m$. The resulting disparity maps D_b and D_m are then filtered and subject to a left-right consistency check (LRC, see section 4.2). Based on the filtered results pixel-wise disparity search ranges $T_b^{min}(\mathbf{x}_b)$, $T_b^{max}(\mathbf{x}_b)$, $T_m^{min}(\mathbf{x}_m)$ and $T_m^{max}(\mathbf{x}_m)$ for the next lower pyramid are computed. The structures T^{max} and T^{min} hold pixel-wise disparity ranges. The complete range per pixel is defined by

$$\begin{aligned} d_{max}(\mathbf{x}) &= D(\mathbf{x}) + T^{max}(\mathbf{x}) \\ d_{min}(\mathbf{x}) &= D(\mathbf{x}) - T^{min}(\mathbf{x}) \end{aligned}. \tag{4.7}$$

Thereby values in T_{max} and T_{min} are bounded such that for each pixel $T^{max}(\mathbf{x}_b)+T^{min}(\mathbf{x}_b) < R$, thus search ranges are ensured not to exceed a predefined range R. The selection of search ranges is a crucial point: on the one hand large ranges increase processing time, memory requirements and ambiguities when matching repetitive texture. On the other hand, if ranges are too narrow reconstruction of small details which are not

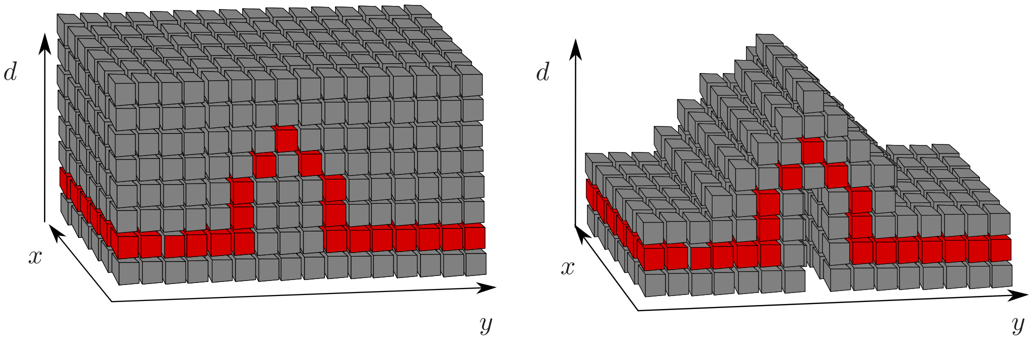


Figure 4.7: Cost structures of classic SGM (left) and tSGM (right). Red cubes represent costs for the true correspondences. Gray cubes mark the costs of potential correspondences, thus the disparity search ranges. Note that disparity search ranges for single pixels differ and search ranges to the right (upper gray cubes) differ from search ranges to the left (lower gray cubes).

mapped in low resolution pyramids and surfaces at disparity continuities are hindered. Our search range computation distinguishes successfully matched and not successfully matched pixels as available from the filtered disparity maps. For pixels which could not be reconstructed (invalid pixels) search ranges for the next higher image pyramid are computed more conservatively assuming that on higher pyramids small details are visible or texture might be beneficial such that disparities can be recovered. Thus, search ranges are only moderately limited in contrast to successfully matched regions. For invalid pixels $\mathbf{x}$ the median value d_{med} of an 41×41 window around the respective pixel in the disparity image D is determined. If at least three valid disparities were found the range $D(\mathbf{x})$ is set to d_{med} and $T^{max}(\mathbf{x}) = \frac{R}{2}$ and $T^{min}(\mathbf{x}) = \frac{R}{2}$. If not at least 3 valid disparities are contained in the window the median value is considered unreliable and $D(\mathbf{x})$ is set to the mean disparity computed over the whole disparity image. As mentioned before search ranges d_{max},d_{min} for valid disparities are limited more aggressively. Therefore minimum and maximum disparities d_{min} and d_{max} in a 7×7 window are determined. If $d_{max} - d_{min}$ does not exceed the maximum range R the search ranges are specified to $t^{max} = d_{max} - D(\mathbf{x}) + 2$ and $t^{min} = D(\mathbf{x}) - d_{min} + 2$. Otherwise, if $d_{max} - d_{min} > R$ the search ranges are scaled according to

$$\begin{aligned} t^{max} &= R\frac{d_{max} - D(\mathbf{x})}{d_{max} - d_{min}} \\ t^{min} &= R\frac{D(\mathbf{x}) - d_{min}}{d_{max} - d_{min}} \end{aligned} \tag{4.8}$$

After all values of T_{max} and T_{min} are computed they are multiplied by two and upscaled to fit the next disparity level. The result is then utilized to restrict search ranges in the matching process of the next higher image pyramid $l - 1$. This iterative process is terminated on completion of matching the full resolution imagery $l = 0$.

By the pixel-wise adaption of disparity search ranges the cubic shape of arrays holding the local costs $C(\mathbf{x}_b, d)$ and $S(\mathbf{x}_b, d)$ is no longer guaranteed (figure 4.7). In disparity space these structures represent a band containing potential disparities of the surface estimated on higher image pyramids. For the computation of C and S and in particular the cost accumulation process a efficient access to elements of the structures has to be guaranteed. In practice all values of the structures C and S are stored subsequently in one dimensional arrays. The first elements of cost strings associated with a base image pixel are accessed using an image providing the respective offsets in the C and S arrays. Furthermore, the algorithm for path accumulation as given in equation (4.5) has to be redesigned. Since cost strings of neighboring pixels may overlap only partly or do not overlap at all, the costs $L_r(\mathbf{x}_b - \mathbf{r}_i, d + k)$ might not exist (see figure 4.5). In this case the bottom or top elements of the neighboring cost string $L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_{min}(\mathbf{x}_b - \mathbf{r}_i))$ and $L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_{max}(\mathbf{x}_b - \mathbf{r}_i))$ are employed. The recursive strategy given in equation (4.5) is enhanced by a case distinction according to

$$\begin{aligned}
&\textbf{if} \quad d > d_{max}(\mathbf{x}_b - \mathbf{r}_i): \\
&\quad \bar{L}_{\mathbf{r}_i}(\mathbf{x}_b, d) = C_{\mathbf{r}_i}(\mathbf{x}_b, d) + L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_{max}(\mathbf{x}_b - \mathbf{r}_i)) \\
&\quad + P_2 - min_k L_r(\mathbf{x}_b - \mathbf{r}_i, k) \\
&\textbf{elseif} \quad d < d_{min}(\mathbf{x}_b - \mathbf{r}_i): \\
&\quad \bar{L}_{\mathbf{r}_i}(\mathbf{x}_b, d) = C_{\mathbf{r}_i}(\mathbf{x}_b, d) + L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d_{min}(\mathbf{x}_b - \mathbf{r}_i)) \\
&\quad + P_2 - min_k L_r(\mathbf{x}_b - \mathbf{r}_i, k) \\
&\textbf{else}: \\
&\quad L_{\mathbf{r}_i}(\mathbf{x}_b, d) = C(\mathbf{x}_b, d) + min(L_r(\mathbf{x}_b - \mathbf{r}_i, d), \\
&\quad L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d-1) + P_1 \\
&\quad L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, d+1) + P_1, \\
&\quad L_{\mathbf{r}_i}(\mathbf{x}_b - \mathbf{r}_i, i) + P_2) \\
&\quad - min_k L_r(\mathbf{x}_b - \mathbf{r}_i, k).
\end{aligned} \tag{4.9}$$

The path accumulation step is the computational most expensive within the matching process. A nice property of the algorithm is that it can be parallelized conveniently, more precisely single paths of the same path direction can be processed in parallel. For this low-level parallelization the parallelization framework OpenMP[OpenMP Architecture Review Board, 2012] was utilized.

Another strategy to reduce processing time is to reduce the passes through the cost structures in the course of cost accumulation. When passing the DSI along the vertical paths not only costs along vertical directions but also along two diagonal paths are accumulated, see figure 4.3. This way the number of passes can be reduced from 8 to 4.

Hierarchical Determination of Mutual Visibility As explained in the previous paragraph disparity images from lower levels are utilized to limit disparity search ranges. Furthermore, low resolution disparity maps are used to identify the scene extend which is commonly observed in the two images. These regions are estimated based on consistent disparities, e.g. disparities not removed by the implemented filters. This information is stored in additional visibility masks $V_b(\mathbf{x}_b)$, $V_m(\mathbf{x}_m)$. Based on these binary maps the stereo matching process is initialized such that non-valid pixels are discarded in completely in the course of dense matching which additionally lowers processing times and memory requirements. The binary visibility maps are initialized using the disparity maps D_b and D_m respectively. For valid disparities $V_b(x, y)$ and $V_m(x, y)$ are set to 1 and to 0 otherwise. Since disparity maps contain spurious elements, V_b and V_m are subject to a filtering process. First, a speckle filter is applied removing connected components possessing only small extend. Thereby neighboring pixels are considered connected if they are valid. Within a second step V_b and V_m are skimmed from left to right, top to bottom and vise versa. Along each scanline the pixels are checked for validity, if a pixel is valid a counter n_v is increased by one. All pixels passed before the counter reaches the threshold t_v are invalidated. The benefit of this filter is discussed in section 4.4.1.

Computation of the Smoothing Parameters P_1 and P_2 The smoothness of the surface is controlled by the penalty parameters P_1 and P_2. Thereby for each pixel $\mathbf{x}$ in the base image $\mathbf{I}_b$ and a pixel in its direct neighborhood $\mathbf{y}$ a penalty $P(d_\mathbf{x}, d_\mathbf{y})$ is computed. Theses functions are called interaction functions. A rather common interaction function in dense stereo for example is the Potts model [Veksler, 2007] given by

$$P_{\mathbf{x},\mathbf{y}} = w_{\mathbf{x},\mathbf{y}} min(1, d_\mathbf{x} - d_\mathbf{y}) \tag{4.10}$$

where $w_{\mathbf{x},\mathbf{y}}$ is weight varying for different pixel pairings. Instead the path aggregation in the SGM method utilizes a linear truncated function

$$P_{\mathbf{x},\mathbf{y}} = min(P_1|d_\mathbf{x} - d_\mathbf{y}|, P_2) \quad with \quad 0 < P1 < P2 \tag{4.11}$$

The penalty parameter P_1 is designed to control the smoothness of the surface for neighboring disparities possessing only small variances. P_2 is introduced to control the penalty for depth discontinuities. For large values of P_2 disparity discontinuities are hindered otherwise discontinuities are tolerated. A typical assumption is that depth discontinuities are indicated by large intensity gradients in the images. Therefore, P_2 takes large values for low absolute intensity gradients between $\mathbf{x}$ and its neighbor $\mathbf{y}$ and small values when the absolute gradient is large. Instead of utilizing intensity gradients in the directions of paths we prefer pixel-wise penalties based on robust edge detection as implemented by the Canny algorithm [Canny, 1986]. The binary response of the canny filter $C(\mathbf{x})$ is then used to compute P_2 for each single pixel as follows.

$$P_2(\mathbf{x}) = \begin{cases} P_{21} - P_{22} & for \quad C(\mathbf{x}) = 1 \\ P_{21} & for \quad C(\mathbf{x}) = 0 \end{cases} \tag{4.12}$$

Thereby it has to be assured that $P_{21} - P_{22} > P_1$. Of course parametrization is dependent on the utilized photo-consistency measures. However, for the Census matching cost the parametrization is extremely robust and we found $P_{21} = 100$, $P_{22} = 99$ and $P_1 = 28$ give reliable results.

Disparity Refinement And Filtering of Disparity Maps After the process of cost accumulation the most probable disparities from the array of accumulated cost $S(\mathbf{x}, d)$ can be derived. This is implemented by a winner-takes-it-all strategy selecting the disparity for a base image pixel $\mathbf{x}$ according to the minimal cost $S(\mathbf{x}, d)$ with $d = [d_{min}, ..., d_{max}]$ in the range defined by $T^{min}(\mathbf{x})$ and $T^{max}(\mathbf{x})$. Since we are operating in discrete disparity space minimal cost define disparities at full integer positions only. Let d_0 be the disparity associated with the minimal cost and let d_1, d_{-1} be the disparities at the neighboring integer positions. A common strategy to estimate sub-pixel disparities is to fit a parabola to the aggregated costs $S_{-1} = S(\mathbf{x}, d_{-1})$, $S_0 = S(\mathbf{x}, d)$ and $S_1 = S(\mathbf{x}, d_1)$. The minimum of the parabola then defines the final sub-pixel disparity. For fitting the parabola the disparities are expressed relative to the disparity d_0 such that $d'_0 = 0$, $d'_{-1} = -1$ and $d'_1 = 1$. The parameters of the parabola $f(d) = ad'^2 + bd' + c$ are given by:

$$c = S_0 \quad b = \frac{S_1 - S_{-1}}{2} \quad a = \frac{S_1 - S_{-1}}{2} - S_0. \tag{4.13}$$

The sub-pixel disparity equates to $d_{sub} = d_0 + d_o$ where the offset is computed as

$$d_o = \frac{-b}{2a} = \frac{1}{\frac{2S_0}{S_1 - S_{-1}} - 1}. \tag{4.14}$$

by claiming $\frac{\partial f}{\partial d} = 0$. If at least one of the elements d_{-1}, d_0, d_{+1} is unavailable sub-pixel estimation is skipped. Unfortunately this technique suffers from an effect called pixel locking, e.g. [Shimizu and Okutomi, 2002], [Xiong and Matthies, 1997]. This interpolation error causes sub-pixel disparities not be equally distributed but biased towards integer valued disparities. As mentioned in section 2.2.2 several strategies to reduce the pixel locking effect were proposed. However, additional time requirements are considerable and therefore we abandon further refinement. Sub-pixel disparity maps are subject to a filter procedure (see figure4.6) to remove spurious elements. First, disparity maps are subject to speckle filter which remove connected components of small extent. Thereby pixels are considered to be connected if they are direct neighbors and their disparity values do not vary more than a certain threshold. After that median filtering with a kernel size of 3 × 3 pixels is applied. So far the complete matching procedure can be carried out in parallel with the roles of base and match image interchanged as shown in figure 4.6. In contrast the last filter technique, the left-right consistency check, requires the disparity maps D_b and D_m from both processing strings. Thereby the disparities from both processes are checked for mutual consistency, in other words it is validated if the estimated depths across two views which correspondent to the same object point are similar. Let $d_b = D_b(\mathbf{x}_b)$ be a disparity in the depth map dedicated to the base image. The floating point coordinates of correspondence in the match image is given by $\mathbf{x}_m = (x + d_b, y)$. Interpolation of the disparities in D_m delivers the disparity $d_m(x + d_b, y)$ which should map back to x_b. Practically we set

$$D(\mathbf{x}_b) = \begin{cases} valid & if \quad |d_b + d_m| < 1 \\ invalid & else \end{cases} \tag{4.15}$$

The left-right consistency check completes the filter procedure and the resultant depth maps D_b and D_m are then used to derive disparity range images T_b^{min}, T_b^{max}, T_m^{min}, T_m^{max} and validity images V_b and V_m as explained in sections 4.2 and 4.2.

Photo Consistency Measures As mentioned before much research on matching costs was conducted in the photogrammetric and computer vision community in the last 3 decades. For dense stereo generally area based photo similarity measures are preferred to feature based measures. The reason is better accuracy and beneficial run time. A short review of common photo consistency measures are given is section 2.1.3. Within the implemented pipeline two consistency measures were implemented: Census [Zabih and Woodfill, 1994] and AD-Census [Sun et al., 2011].

The key strength of the Census based correlation is its robustness against radiometric variances across image pairs. Beside this it can be computed effectively. For the evaluation function, the so called Hamming distance, even build-in compiler commands are available. Census correlation is a window based similarity measure. Let $\mathbf{x}_{0,0}$ be a pixel to be evaluated and $\mathbf{x}_{i,j}$ be a pixels in its surrounding window. The intensities of each window pixel is compared to the central intensity and a binary response $r_{i,j}$ is computed according to

$$r_{i,j} = \begin{cases} 1 & if \quad \mathbf{x}_{i,j} > \mathbf{x}_{0,0} \\ 0 & else \end{cases} \tag{4.16}$$

For $i = [-\frac{N-1}{2}..\frac{M-1}{2}]$ this results in a binary vector of the length MN. Typically the window dimensions N and M are chosen such that MN does not exceed 64 bits which can be conveniently stored by standard data types, e.g $M = 7$ $N = 9$. The actual similarity of two pixels is then defined by the Hamming distance of their binary vectors. The distance is defined by

$$C(\mathbf{x}_{0,0}, \mathbf{y}_{0,0}) = \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=-\frac{M-1}{2}}^{\frac{M-1}{2}} (r_{i,j} \text{ XNOR } q_{i,j}) \tag{4.17}$$

where $q_{i,j}$ is the bit string associated with the second pixel, computed according equation 4.16. In case of AD census equation 4.17 is enhanced by the weighted absolute intensity differences of center pixels

$$C(\mathbf{x}_{0,0}, \mathbf{y}_{0,0}) = \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=-\frac{M-1}{2}}^{\frac{M-1}{2}} (r_{i,j} \text{ XNOR } q_{i,j}) + w|\mathbf{x}_{0,0} - \mathbf{y}_{0,0}|. \tag{4.18}$$

Thereby w is a weight to controls the impact of the absolute difference term. As mentioned before we found the Census and AD-Census correlation to yield solid results in terms of speed, precision, reconstruction density and robustness against parametrization. In fact for the standard Census correlation we use same parameters for all datasets and type of imagery.

Adaptive Tiled Processing As mentioned before one of the advantages of the SGM method is the low memory footprint. Nevertheless, large frame stereo pairs can not be processed as a whole due to memory demands. To overcome this problem, within the classical approach images are split into quadratic tiles which then are matched. Let $\mathbf{J}_b^{x,y}$ be a tile of $\mathbf{I}_b$ with the indexes x,y denoting the location in $\mathbf{I}_b$. After extending the tile borders by a certain overlap the single tiles $\mathbf{J}_b^{x,y}$ are subsequently matched independent from each other and the respective disparity tiles are fused on completion. Of course for tiled based stereo matching the correspondent image region $\mathbf{J}_m(\mathbf{J}_b^{x,y})$ in the match image $\mathbf{I}_m$ has to be identified. This can be done based on the terrain height, which presumes some pre-knowledge of the scene or results from previous matching iterations which might lack robustness. Therefore, determination of the region $\mathbf{J}_m(\mathbf{J}_b^{x,y})$ has to be designed conservatively. In any case a certain processing overhead is involved. The reduced memory demands by our approach allows for row-based processing. Thereby the image is only split in vertical tiles covering the whole image width. The advantage is that now the tile selection process is not required anymore. Moreover, processing overhead as a result from overlaps in x directions are eliminated as well as processing overhead due to conservative estimations of $\mathbf{J}_m$ can be avoided. Furthermore, we select y dimensions of tiles for each tile separately. Based on the range structures $\mathbf{T}^{max}$ and $\mathbf{T}^{min}$ and the offset image available from the previous matching iteration the required memory for C and S and additional helper structures can be computed row-wise. Tile borders in vertical directions are then selected such that the system memory or a user defined memory limit is not exceeded. By optimizing tile dimensions the number of tiles and therefore the computation in overlapping regions is minimized. Furthermore, the possibility of user defined memory limitations is an important functionality when utilizing systems where multiple instances of the program are executed on a single processor.

4.3 Multi Baseline Triangulation

As outlined in section 3 within our reconstruction solution each base image $\mathbf{I}_b$ is stereo matched against a set of match images $\mathbf{I}_{m,j} \subseteq \mathcal{N}_b$. In a final step redundant depth estimates across the disparity images are identified and fused in order to filter outliers and increase the precision of final depth maps. Assume we generated a set of disparity maps $D_b^{r,j}$ as a result of matching rectified versions of the base image $\mathbf{I}_b^{r,j}$ against rectified versions of match images $\mathbf{I}_{m,j}^r$. In this section we describe how we compute the final depths for each pixel $\mathbf{x}_b$ in the non-rectified base image by fusing redundant observations encoded by the disparity images $D_b^{r,j}$. In a first step we link pixel coordinates of the base image to pixel coordinates in the match images. This process is referred to as correspondence linking and discussed in section 4.3.1. Each of these links implies a depth estimate. Redundant depth estimates are then checked for geometric consistency and non-consistent links are invalidated as discussed in detail in section 4.3.2. The remainder is used for multi-baseline forward intersection (section 4.3.3) leading to the final depths of the base image pixel $\mathbf{x}_b$. The concept of correspondence linking was also employed in [Koch et al., 1998] and [Pollefeys et al., 1998], however we utilize a different methodology for subsequent outlier elimination and multi-baseline triangulation.

4.3.1 Correspondence Linking

The goal of the correspondence linking step is to identify the set of image coordinates $\mathbf{x}_m$ in the match images which are implied by the redundant disparity estimates correspondent to a pixel $\mathbf{x}_b$ in the base image $\mathbf{I}_b$. In the course of image rectification a transformation Φ was computed to transfer base image pixels to their rectified versions $\mathbf{I}_b$, e.g $\mathbf{x}_b^r = \Phi_b(\mathbf{x}_b)$ (see equation 4.1). Because dense stereo was carried out on rectified versions of $\mathbf{I}_b$ and $\mathbf{I}_m$, the rectified base image coordinates $\mathbf{x}_b^r$ are required to access the disparities $D_b^r(\mathbf{x}_b^r)$. To establish the links between base images and its rectified versions the transformation Φ is utilized. Note that $\mathbf{x}_b^r$ is in generally not located at integer pixel coordinates and the disparity value has to be derived using interpolation. The link of a rectified base image pixel $\mathbf{x}_b^r = (x_b^r, y_b^r)$ to its correspondence in the match image can then be established by $\mathbf{x}_m^r = (x_b^r + D_b^r(\mathbf{x}_b^r), y_b^r)$. Summarized, the complete link from base to match image pixels can be obtained following

$$\mathbf{x}_b^r = (x_b^r, y_b^r) = \Phi_b(\mathbf{x}_b) \tag{4.19}$$

$$\mathbf{x}_m = \Phi_m^{-1}(x_b^r + D_b^r(\mathbf{x}_b^r), y_b^r)). \tag{4.20}$$

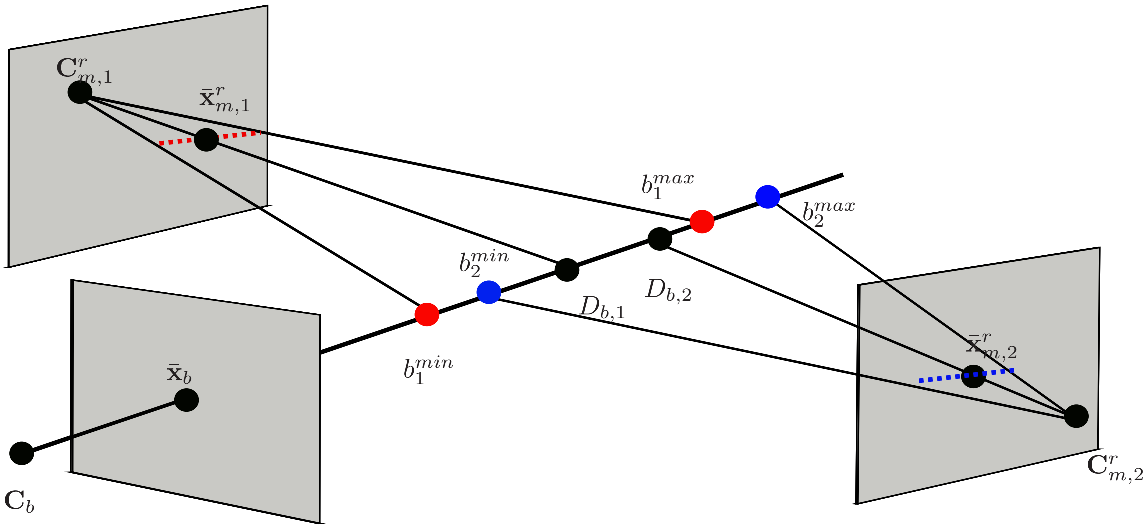


Figure 4.8: Two observations $\bar{\mathbf{x}}_{m,1}$,$\bar{\mathbf{x}}_{m,2}$ in the rectified match images imply the depths $D_{b,1}$, $D_{b,2}$ on the non-rectified base image ray $\bar{\mathbf{x}}_b$. Confidence intervals for disparities along epipolar lines (blue and red dotted lines) induce a range on the base image ray $[b_n^{min}, b_n^{max}]$. If these ranges overlap disparity estimations are considered consistent.

Within the correspondence linking module Φ_b as well as $(\Phi_m)^{-1}$ are recalculated dependent on the type of rectification. For the homography-based rectification types Φ_b and Φ_m^{-1} are specified by the homographies $\mathbf{H}_b$ and $\mathbf{H}_m$. These homographies transferring coordinates from base or match images to their rectified versions can be easily calculated according to

$$\mathbf{H}_b = \mathbf{K}_b^r \mathbf{R}_b^r (\mathbf{K}_b \mathbf{R}_b)^{-1} \tag{4.21}$$

and analogously

$$\mathbf{H}_m = \mathbf{K}_m^r \mathbf{R}_m^r (\mathbf{K}_m \mathbf{R}_m)^{-1}. \tag{4.22}$$

For polar rectified image pairs Φ_b and Φ_m^{-1} are computed based on arrays encoding the angle increments between epipolar lines and arrays encoding the distance from epipoles to the image borders. As mentioned before we recompute these arrays instead to avoid input/output operations.

4.3.2 Geometric Consistency Filters

Similar to the left-right consistency check carried out in dense stereo the redundant observations established by correspondence linking are now checked for geometric consistency. Each of the links define a depth on the base image ray induced by $\mathbf{x}_b$. These pair-wise depths are subject to our filter algorithm and their derivation depends on the type of the utilized rectification type. Depth computation is explained in the following two paragraphs prior to the discussion of the actual consistency check.

Depth From Stereo For Homography-Based Rectification For homography-based rectified image pairs rotations are equal and viewing directions are orthogonal to the baseline, therefore $\mathbf{R}_3^r(\mathbf{C}_b^r - \mathbf{C}_m^r) = 0$. Thereby $\mathbf{R}_3^r$ is the third row of the rotation matrix defining the viewing direction and $\mathbf{C}_b^r$,$\mathbf{C}_m^r$ are the camera positions. Furthermore focal lengths are equal. Stereo pairs with this properties are referred to as *stereo normal case* [Kraus, 1994] and the depth with respect to the camera viewing directions can be easily computed as

$$Z_b^r = \frac{Bf}{d_b^r}. \tag{4.23}$$

Thereby B denotes the baseline $B = \|\mathbf{C}_b^r - \mathbf{C}_m^r\|_2$, f is the common focal length and d_b^r represents the disparity. Let $\mathbf{x}_b = (x_b, y_b, 1)$ represent the homogeneous base image coordinates and $\bar{\mathbf{x}}_b^r = (\bar{x}_b^r, \bar{y}_b^r, 1) \sim (\mathbf{K}_b^r)^{-1}\mathbf{x}_b^r$ be the normalized coordinates defining a the base image ray with respect to the base image coordinate system. Thereby $\sim$ denotes equality up to scale. Then the depth along this ray can be computed as

$$D_b^r = \frac{B\sqrt{(\bar{x}_b)^2 + (\bar{y}_b)^2 + 1}}{d_b^r}. \tag{4.24}$$

The corresponding 3D coordinate with respect to the object space coordinate system is then computed as $\mathbf{C}_b^r + D_b^r \mathbf{R}_b^\top \bar{\mathbf{x}}_b^r$. Note that not the full link from $\mathbf{x}_b$ to $\mathbf{x}_m$ was used but the links from rectified base and match images only. Furthermore, it is important to note that the depth with respect to the rectified base image D_b^r equals the depth D_b of the ray along $\bar{\mathbf{x}}_b$ in the non-rectified base image. This results from the fact that a homography map a point from the original image plane to a rectified image plane without changing perspective centers. This way directions of the rays vary with respect to the single image coordinate systems but equal with respect to the world coordinate system. For the rectification using [Fusiello et al., 2000] this can be easily seen by reformulation of equation 8.16 to

$$\mathbf{R'}^\top \mathbf{K'}^{-1} \mathbf{x'}_b = \mathbf{R}_b^\top \mathbf{K}_b^{-1} \mathbf{x}_b \tag{4.25}$$

$$\mathbf{R'}^\top \mathbf{x'}_b^n = \mathbf{R}_b^\top \mathbf{x}_b^n \tag{4.26}$$

$$\mathbf{x}_w^n = \mathbf{x}_w^n \tag{4.27}$$

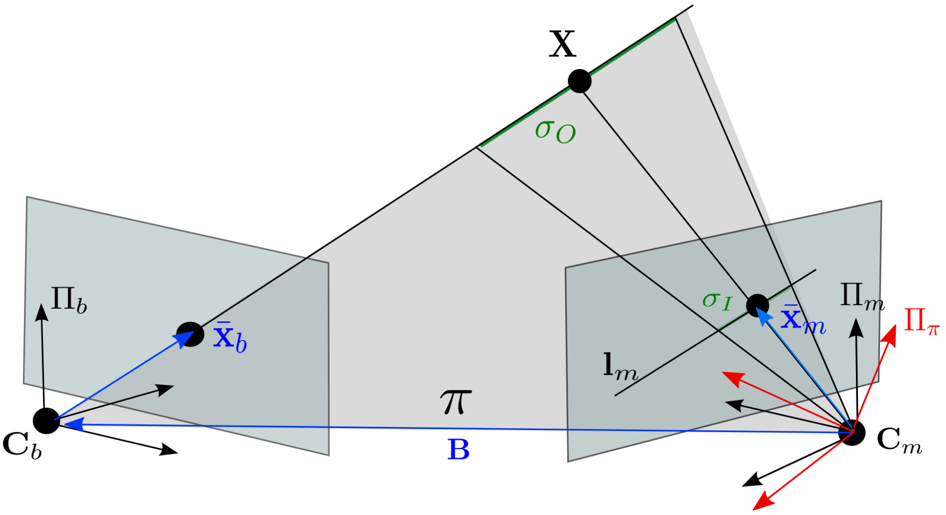


Figure 4.9: Transformation of the forward intersection problem to two dimensions. The problem is reformulated with respect to the new coordinate system Π_π. Since $\bar{\mathbf{x}}_b$, $\bar{\mathbf{x}}_m$ and the object point $\mathbf{X}$ are located in the epipolar plane π, the depth along $\bar{\mathbf{x}}_b$ can be derived using a triangle equation. Similarly a uncertainty interval σ_I can be propagated to the base image ray $\bar{\mathbf{x}}_b$ resulting in the uncertainty interval σ_O.

Depth From Stereo For Arbitrary Rectification In contrast, when working with polar-rectified image pairs, the stereo normal case is not given. Therefore we have to compute the depths from single stereo models in a different manner. Let $\mathbf{I}_b$ and $\mathbf{I}_m$ be pair of calibrated base images and $\bar{\mathbf{x}}_b = (\bar{x}_b, \bar{y}_b, 1)$ $\bar{\mathbf{x}}_m = (\bar{x}_m, \bar{y}_m, 1)$ the respective normalized coordinates derived by correspondence linking. To derive the pair-wise depth, the intersection problem is reduced to 2 dimensions by reformulation using the epipolar plane π as depicted in figure 5.12. Therefore, a coordinate system Π_π is defined with the origin located in the optical center C_m. The x-axis of the new coordinate system Π_π is defined by the epipolar line $\mathbf{l}_m$ in $\mathbf{I}_m$, the z-axis is the normal vector $\mathbf{n}_\pi$ of the epipolar plane π. Both entities are defined with respect to the camera coordinate system

Π_m of the camera m. The y-axis is then constructed perpendicular to $\mathbf{n}_\pi$ and $\mathbf{l}_m$. To compute the depth we utilize the triangle equation

$$D_b \bar{\mathbf{x}}_b - \alpha \bar{\mathbf{x}}_m + \mathbf{B} = \mathbf{0} \tag{4.28}$$

with the base line vector $\mathbf{B} = (B_x, B_y)$ and the scale factor α. Thereby all entities are expressed with respect to Π_π. The depth D_b corresponding to the 3D point can then be computed by requiring the two dimensional vector equation of triangles to be zero, which leads to

$$D_b = \frac{B_y \bar{x}_m - B_x \bar{y}_m}{\bar{y}_m \bar{x}_b - \bar{y}_b \bar{x}_m}. \tag{4.29}$$

Note that this formulation is valid for perspective stereo pairs and independent from the type of rectification. This allows us to simultaneously triangulate depths from disparity maps generated by different rectification types. Furthermore, depths $D_{b,n}$ are computed in the direction of $\bar{\mathbf{x}}_b$ for each of the N redundant links and therefore can be directly related.

Geometric Consistency Filtering To check for geometric consistency, subsequent to correspondence linking for each link the redundant depth D_n for $n = 1, ..., N$ is calculated. Similar to the redundant observations an uncertainty interval σ_i along the epipolar line is propagated to the base image ray using equation 4.24 and 4.29 respectively. This results in N uncertainty intervals $[b_n^{min}, b_n^{max}]$ on the base image denoted by sigma $\sigma_{O,n}$, see figure 4.8. If the single intervals overlap the correspondent observations are considered to be consistent. The cluster which contains the largest number of consistent measurements defines the final set of valid observations used within the forward intersection process. If clusters possess the same number of entities, the set possessing the smallest average ray intersection angle

$$\beta = \frac{1}{N} \sum_n \angle(\bar{\mathbf{x}}_{b,n}, \mathbf{C}_n, \bar{\mathbf{x}}_{m,n}) \tag{4.30}$$

defines the valid set of observations. Thereby we argue that for small intersection angles image similarity is assumed to be larger and matching yields more reliable results, although possessing reduced precision. In practice we also specify a number of minimal consistent observations t_{min}. If the critical number of consistent pair-wise depths is not reached all observations are discarded.

4.3.3 Multi-Baseline Forward Intersecion

Once the set of N geometric consistent observations are known they are utilized for a multi baseline forward intersection. For this task [Hartley and Zisserman, 2004] propose to solve the linear system of equations of the form $\mathbf{AX} = \mathbf{0}$ with

$$\mathbf{A} = \begin{pmatrix} x_0 \mathbf{p}_0^{3\top} - \mathbf{p}_0^{1\top} \\ y_0 \mathbf{p}_0^{3\top} - \mathbf{p}_0^{2\top} \\ x_1 \mathbf{p}_1^{3\top} - \mathbf{p}_1^{1\top} \\ y_1 \mathbf{p}_1^{3\top} - \mathbf{p}_1^{2\top} \\ \cdots \\ x_n \mathbf{p}_1^{3\top} - \mathbf{p}_n^{1\top} \\ y_n \mathbf{p}_1^{3\top} - \mathbf{p}_n^{2\top} \end{pmatrix}. \tag{4.31}$$

Thereby $\mathbf{p}_n^{i\top}$ denotes the transposed of the i-th row of the projection matrix correspondent to the camera n. For each image two rows are added to $\mathbf{A}$. Solving the system of equations the homogeneous object point $\mathbf{X}$ can be derived. A drawback of this approach is that only an algebraic error with no geometric meaning is minimized. Moreover, we would have to solve the system for each pixel which is computational expensive. Instead we propose two alternative approaches for the derivation of multi-baseline forward intersection.

Depth From Multi-Baseline Forward Intersection for Homography-Based Rectification For the rectified images based on homographies we can formulate the forward intersection problem minimizing the reprojection error along the epipolar line using a direct solution. The reprojection error is adequate error metric since it is modeling the different geometric properties of ray intersections as variances in scale and intersection angles by design. We formulate the reprojection error along the epipolar line as

$$\sum_{n=0}^{N}(x_n - \hat{x}_n)^2 = min \tag{4.32}$$

or with $x_n = x_b + d_n$ and $\hat{x}_n = x_b + \hat{d}_n$

$$\sum_{n=0}^{N}(d_n - \hat{d}_n)^2 = min. \tag{4.33}$$

As can be seen from equation 4.24 for each model the depth is related to the disparity

$$d_n = \frac{B_n\sqrt{(\bar{x}_{b,n})^2 + (\bar{y}_{b,n})^2 + 1}}{D_n} := \frac{a_n}{D_n} \tag{4.34}$$

Substitution and the fact that the depths D_n can be directly related across single stereo models leads to

$$\sum_{n=0}^{N}(d_n - \frac{a_n}{\hat{D}_n})^2 = \sum_{n=0}^{N}(d_n - \frac{a_n}{\hat{D}})^2 = min. \tag{4.35}$$

Derivation with respect to $\hat{D}$ and equating to zero the direct solution minimizing the reprojection error can be obtained by

$$\hat{D} = \frac{\sum_{n=0}^{N} a_n^2}{\sum_{n=0}^{N} d_n a_n}. \tag{4.36}$$

Assuming a disparity uncertainty σ_d the uncertainty of the depth σ_D is derived by error propagation resulting in

$$\sigma_D = \sigma_d\sqrt{\sum_{m=0}^{N}\left(\frac{(\sum_{n=0}^{N} a_n^2)(-a_m)}{(\sum_{n=0}^{N} d_n a_n)^2}\right)^2}. \tag{4.37}$$

Depth From Multi-Baseline Forward Intersection for Arbitrary Rectification Minimization of the reprojection error for polar rectified imagery unfortunately can not be formulated in a direct way. However, minimization could be carried out using some iterative approach, for example Gauss Newton. Instead we minimize the error in object space. Thereby single depth observations D_n are weighted according their propagated accuracy intervals $\sigma_{O,n}$ as calculated in section 4.3.2. The minimization functional is given by

$$\sum_{n=0}^{N}\frac{1}{\sigma_{O,n}}(\hat{D} - D_n)^2 \tag{4.38}$$

Derivation and equating to zero leads to

$$\hat{D} = \frac{\sum_{n=0}^{N}\frac{D_n}{\sigma_{O,n}}}{\sum_{n=0}^{N}\frac{1}{\sigma_{O,n}}}. \tag{4.39}$$

4.3.4 Example for Multi-Baseline Stereo

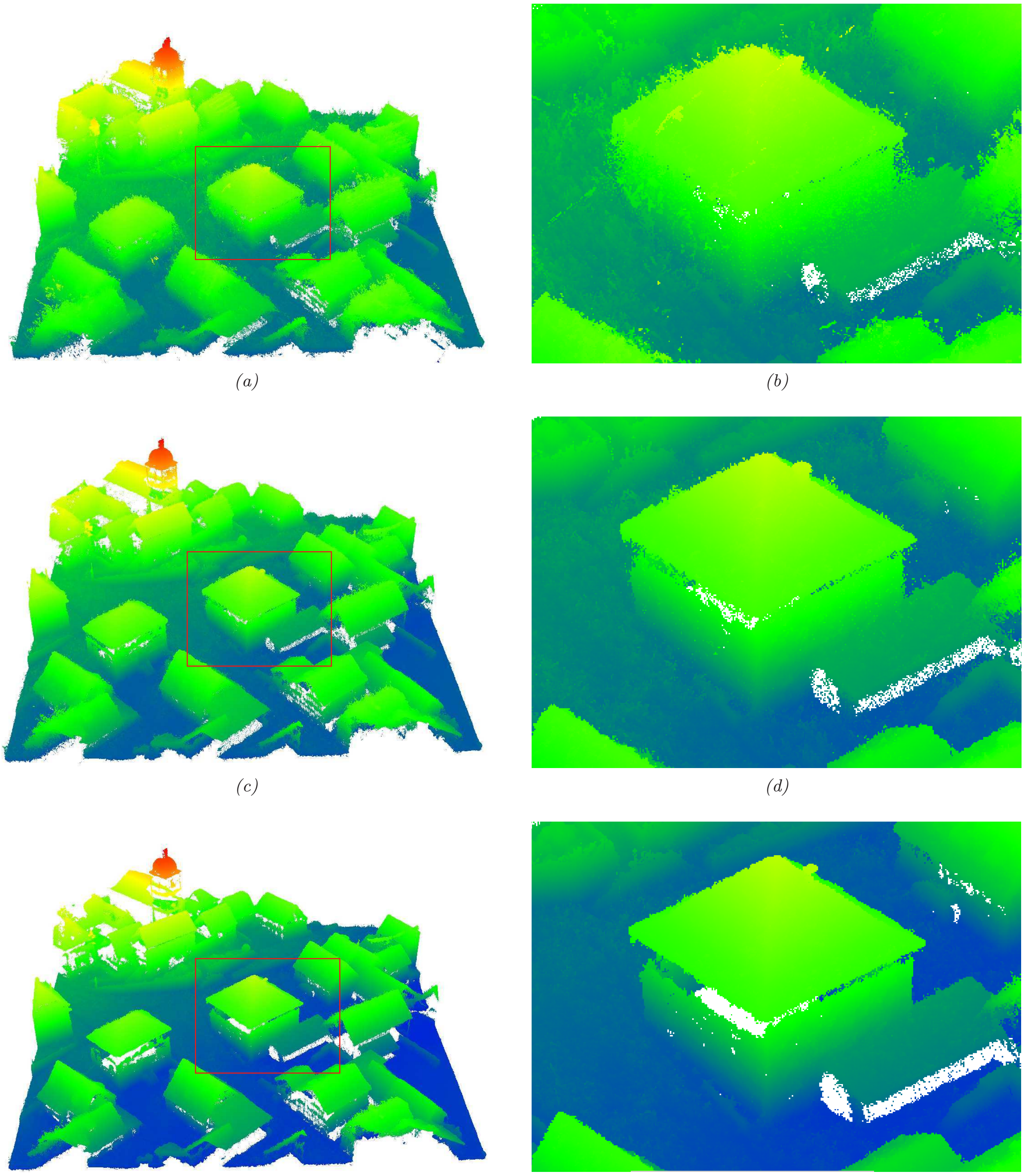
In this section we show the benefits of the proposed algorithms for a set of rather challenging UAV images. In figure 4.10 example point clouds generated using our multi-baseline approach for different values of t_{min} are depicted. In this example UAV imagery was extracted from a video stream. Compared to large-frame airborne camera systems signal-to-noise ratios are rather low and due to the instability and fast movements of the platform possess considerable motion blur. As a result image orientation is less accurate and the computed disparities generally provide higher noise levels and a larger number of outliers. We computed point clouds using different thresholds t_{min} for the minimal number of geometric consistent observations. For a central part of the image block, all points clouds resultant from the triangulation process were merged in object space. Then the number of blunders n_e above the church top and below the ground was recorded and blunders were removed for visualization purposes. Figure 4.10a depicts points generated from pure stereo matching ($t_{min} = 1$). The resulting cloud possesses a large number of gross outliers $n_e = 20141$ as well as a large noise level. By claiming geometric consistency of at least 2 consistent depth estimates per point ($t_{min} = 2$) the number of blunders could be drastically reduced ($n_e = 18$) and even completely eliminated for $t_{min} = 3$. Furthermore, noise levels of the generated points drastically decrease. For a quantitative evaluation of correspondence linking, filtering an forward intersection see section 4.4.2.

4.4 Results

4.4.1 Comparision of classical SGM and tSGM

Within this section the performance of the SGM and tSGM algorithms regarding speed and memory demands are compared. Furthermore differences of the resultant disparities are evaluated. Therefore two sets of rectified image pairs were rectified and matched. The first image pair consists of two 2298×2290 sub-tiles cropped from two large format aerial frames (figure 4.11c). Matching was carried out on full resolution imagery. For the SGM solution some pre-knowledge was incorporated by specifying a constant disparity search range covering exactly all prevalent disparities in the scene. The resulting disparity image generated by SGM was calculated in 44 seconds (PentiumR dual core, 2.6 GHz) and is shown in figure 4.11a. The maximal memory consumption amounted 2.6GB. The parallax image derived by tSGM was computed in 30 seconds and is displayed in figure 4.11b. A visualization of dynamic search ranges for all subsequent pyramid levels is displayed in figure 4.12. Due to the reduced size of the structures used for cost computation and cost accumulation memory consumption of tSGM was reduced by 68.2% to 0.8GB. For same reasons execution time was reduced by 31.8%. Note that for the selected aerial scenario minimal and maximal prevalent disparities do not heavily vary and cube structures are comparable small. For scenes inducing larger variances in depth. e.g undulating terrain as in mountainous regions, memory demands can be reduced by multiples. Similar variances in depth occur for close-range or UAV scenarios. To demonstrate this a second test including two images of the Fountain data set [Strecha et al., 2008] were rectified and matched. Matching using the classical approach was carried out in 65 seconds (I7 quad core, 3.4 GHz). The maximal memory consumption amounted 21.1 GB. The time for matching using the tSGM solution could be reduced by 89.3% to 6.88 seconds. The memory consumption could be reduced by 93.8% to 1.3 GB which enables processing on standard computers. However, in all tests SGM was calculated using the same core algorithms for cost computation and aggregation as used for tSGM. Assuming regular cubic structures, before mentioned operations could be designed and executed more efficiently for the classical SGM approach which could lead to lower processing times. However, optimal disparity search ranges were selected manually which is not feasible for standard processing cycles.

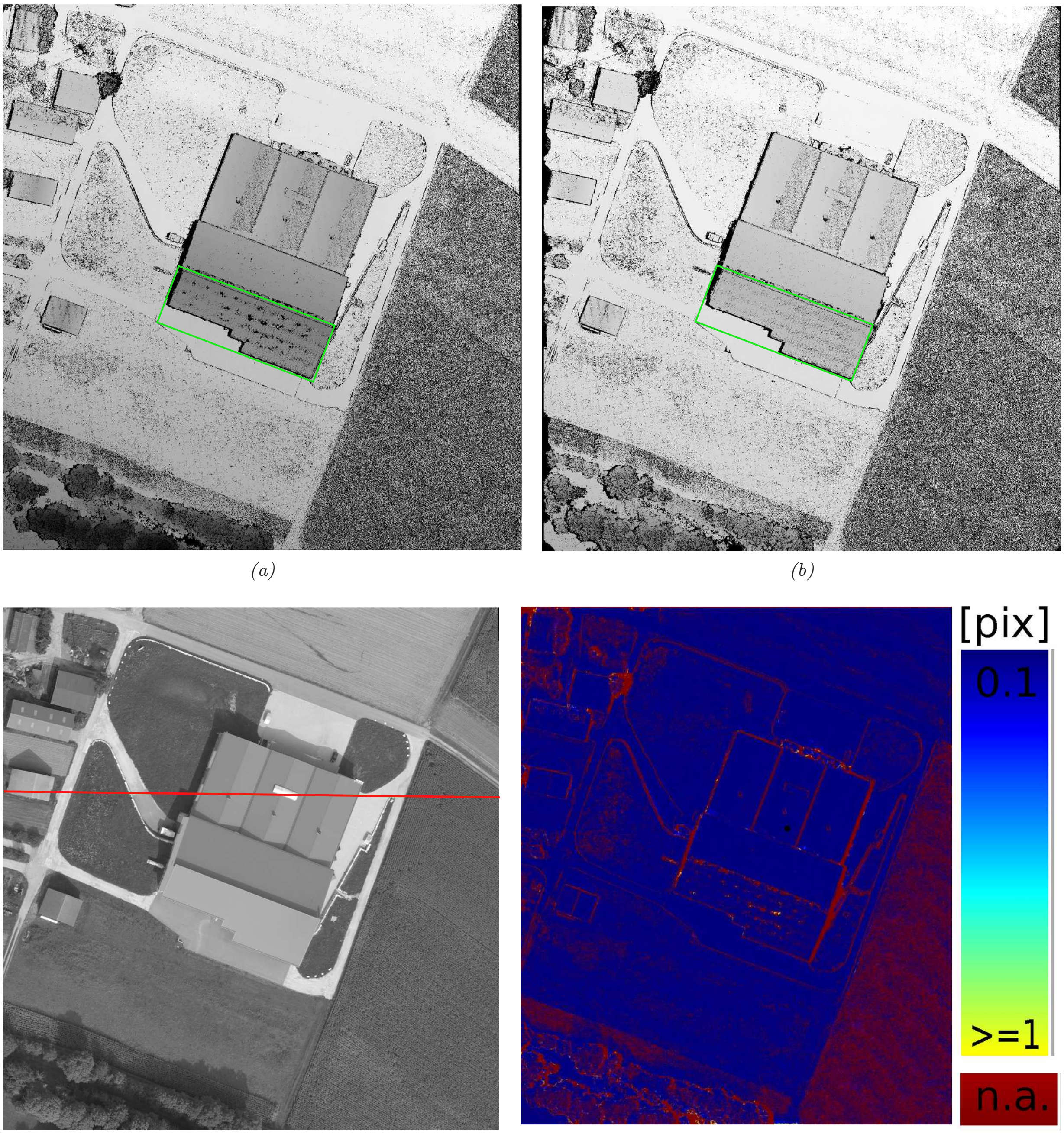
Theoretically tSGM hinders reconstruction of largely undulating structures represented by only few pixels in the images. Small pixel patches might not be passed to lower pyramid levels due to resolution reduction and smoothing. Therefore the predicted search range in the next higher resolution pyramid might not contain the correct disparities and reconstruction for these objects might fail. However, in many data sets the surfaces are captured with various angles. Structures therefore might be represented by a larger number


(a) (b)

(c) (d)

(e) (f)

Figure 4.10: (a): Visualization Point cloud t_{min}=1. (b): Point cloud t_{min}=2, (c): Point cloud t_{min}=3

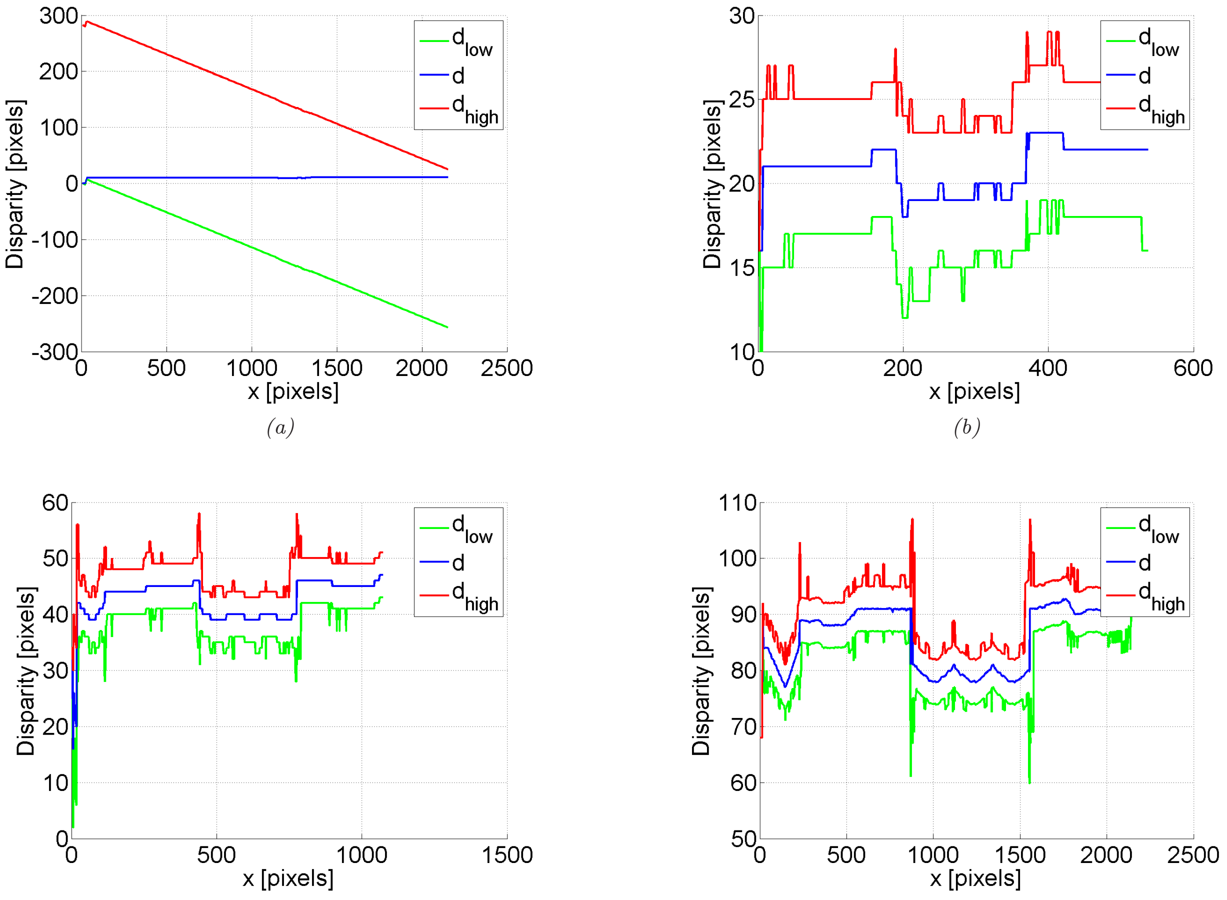


(a) *(b)*

(c) *(d)*

Figure 4.11: Evaluation of deth maps for dense stereo using a snippet cropped from large frame airborne data. (a): Disparity map of classic SGM approach. (b): Disparity map of tSGM, (c): Original image (d): Absolute differences in disparity maps (a) and (b)

of pixels in additional views, which then enables successful reconstruction. Moreover, the proposed tSGM algorithm provides beneficial reconstruction of low textured objects and objects possessing repetitive texture as the roof in the airborne example (figures 4.11a and 4.11b, green boxes). High frequencies patterns are not passed to lower levels which supports a more robust parallax estimation because of reduced ambiguities. In



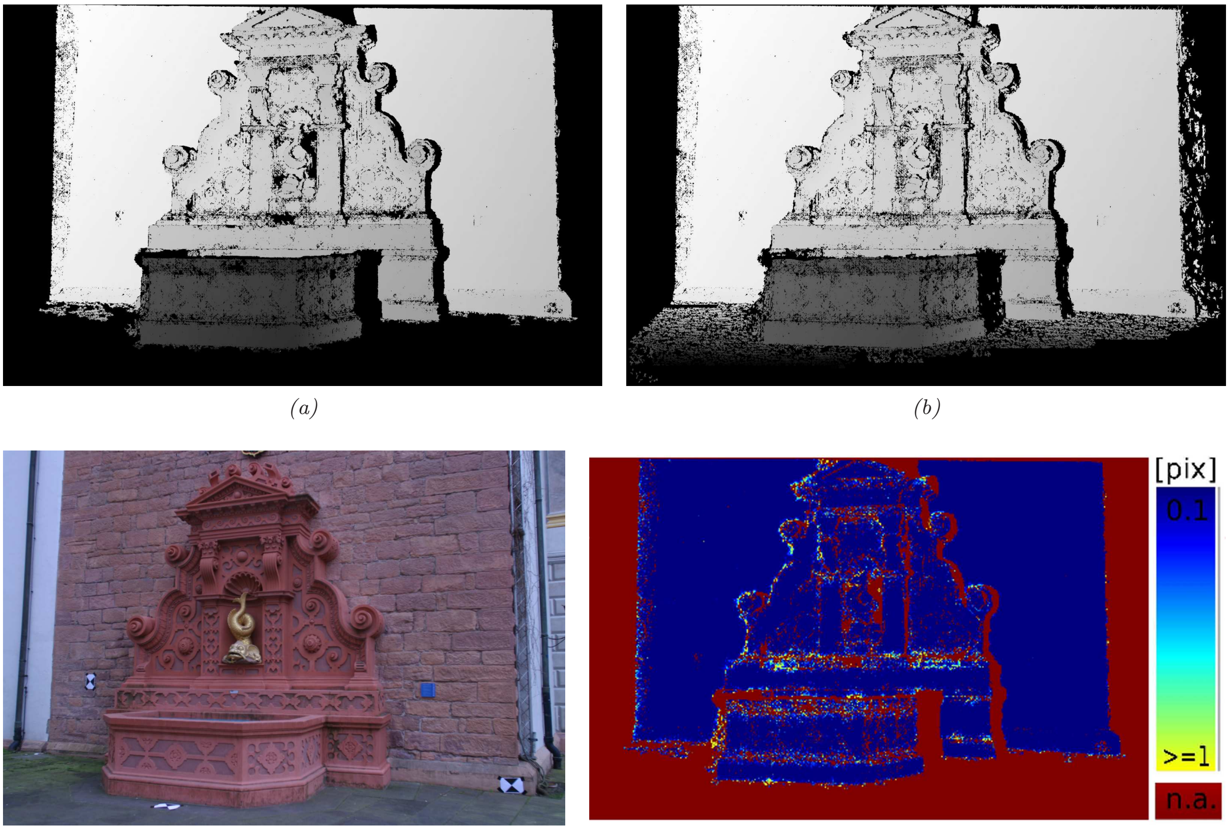
(a) *(b)* *(c)* *(d)*

Figure 4.12: Visualization of disparity search ranges of pyramid levels 3-0 (a-d) for the airborne stereo pair shown figure 4.11. Blue line marks the estimated / interpolated disparity from previous pyramid level. Green and red mark the disparity search range for current level.

subsequent levels ambiguities are diminished due to the reduced search range which leads to a reduction of mismatches. The same observation holds for image parts possessing weak texture and larger differences in appearance as the ground in the fountain data set (figures 4.13a and 4.13b). When matching low pyramid levels the appearance is more similar and Census matching costs are more distinctive since a larger area in object space is captured by the 9×7 correlation window. As before, disparities are propagated to subsequent levels and ambiguities are resolved by the limited disparity search range. This leads to a higher completeness of the disparity maps. Figures 4.13d and 4.13d depict the absolute differences of parallax images produced by the two approaches. The gross of differences regarding successfully reconstructed pixels are close or below 0.1 pixels. Larger differences occur at depth jumps, however the total amount is low and only few differences exceed the 1 pixel limit.

Figure 4.12 depicts the disparity search ranges for the the airborne stereo pair along the red line displayed in figure 4.11c. On the lowest level disparity search ranges are initialized in the complete image range. Subsequently the ranges are narrowed. It can be observed that in contrast to flat areas at large undulations search ranges are increased enabling robust reconstruction at depth jumps.

The surplus of identifying the commonly seen region across stereo pairs as described in section 4.2 is shown



(a) *(b)*

(c) *(d)*

Figure 4.13: (a): Disparity map of classic SGM approach. (b): Disparity map of tSGM, (c): Original image (d): Absolute differences in disparity maps (a) and (b)

by matching a airborne stereo pair with and without the implemented detection filters. Figure 4.14 depicts the stereo pairs whereas figures encoding the information of areas being identified to cover the commonly seen scene extend. Black pixels mark areas which are assumed not to be seen in both of the views and which therefore are discarded within the matching process. Gray and white pixels are assumed to be seen in both images and define the pixels for which correspondences search is carried out. In contrast to gray pixels, white areas mark pixels for which a correspondence was found. The filters detect common regions rather reliable. Processing times using tSGM amounts 114.5 seconds with mutual region filters being deactivated and 60.9 seconds utilizing filters. Beside reducing the number of mismatches the additional time reduction for this example amounts 46.8%.

4.4.2 Evaluation of Multi-Baseline Stereo

For evaluation purposes we use the Ettlingen fountain data set as introduced in [Strecha et al., 2008]. The data set consists of 11 oriented images as displayed in figure 4.15. Additionally ground truth data is available which was acquired using LiDAR device (Zoller+Froehlich IMAGER 5003). According to the publication observations possess standard deviations of about 1.5mm. From the recorded LiDAR points a triangle mesh was generated utilizing Poisson reconstruction [Kazhdan et al., 2006]. To our knowledge this is the only publicly available MVS benchmark data set providing both, oriented imagery and ground truth. For our evaluation we generate a virtual depth map D_{gt} for each image by constructing a virtual view using the respective orientation. Thereby the ray for each pixel is intersected with the triangle mesh and the depth

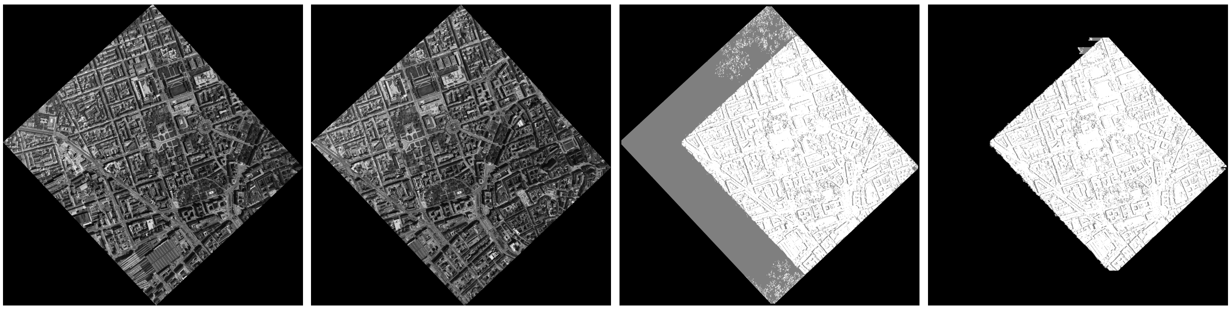


Figure 4.14: Visualization of filter results identifying the commonly observed scene extend. Two left images depict original images. Two right images depict areas (gray and white) for which correspondence search is carried out. Commonly seen areas are detected robustly by our approach (most right image).

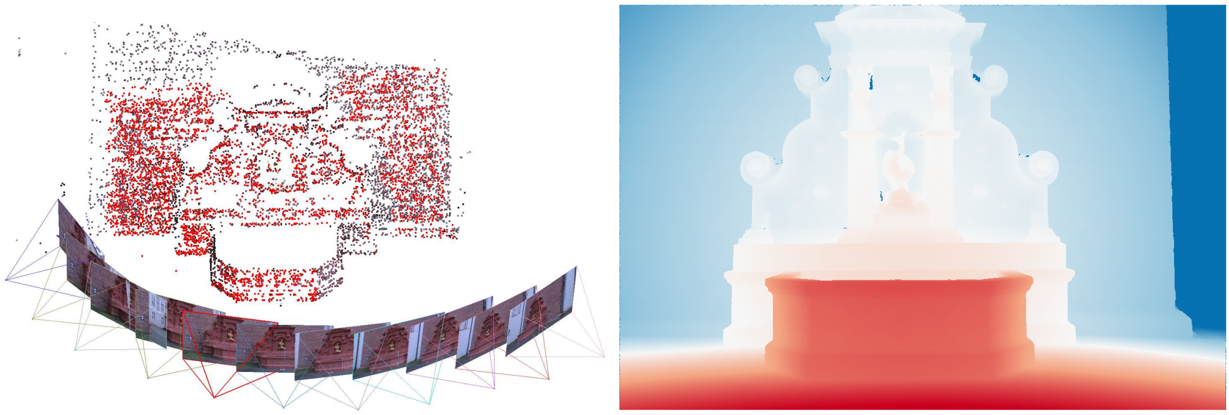


Figure 4.15: Left: Camera configuration of the fountain data set. Right: Depth image generated from ground truth triangle mesh. Dark blue areas and speckles are due to missing data in the ground truth data.

is recorded in D_{gt}. An example is depicted in figure 4.15. For each pixel which is valid in both, D_{gt} and the depth map D_r generated by our multi view approach the differences are evaluated. We employ statistics including the average difference δ, the standard deviation σ, the root mean squared error rms and the percentage of matched pixels with respect to the number of all image pixels. More precisely, mean differences are computed as

$$\delta = \frac{1}{XY} \sum_{x=0}^{X} \sum_{y=0}^{Y} D_r(x,y) - D_{gt}(x,y). \tag{4.40}$$

The standard deviation is computed as

$$\sigma = \sqrt{\frac{1}{XY-1} \sum_{x=0}^{X} \sum_{y=0}^{Y} (D_r(x,y) - D_{gt}(x,y) - \delta)^2}. \tag{4.41}$$

The root mean squared error is defined as

$$rms = \sqrt{\frac{1}{XY-1} \sum_{x=0}^{X} \sum_{y=0}^{Y} (D_r(x,y) - D_{gt}(x,y))^2}. \tag{4.42}$$

In order to remove outliers from the evaluation same characteristics are recomputed after removing values possessing a $\delta > 3\sigma$. To get a better feeling of the characteristics we express the entities in equations 4.40-

4.42 relative to the local pixel footprint. Therefore the differences $D_r(x,y) - D_{gt}(x,y)$ are divided by the local GSD which can be calculated according to

$$GSD(x,y) = D_{gt}(x,y)/f \tag{4.43}$$

where f is the focal length in pixels units. Note that this modification also implies an equal weighting of differences of points which possess different distances to the image plane. However, results of metric evaluations can be found in the appendix 8.3.

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,2}(F)$	-2.36	10.36	10.62	63.77	-1.70	4.35	4.67	63.04
$\mathbf{I}_{b,2}(L)$	-2.27	10.63	10.87	60.82	-1.67	4.35	4.66	60.13
$\mathbf{I}_{b,2}(P)$	-1.93	10.72	10.90	56.98	-1.33	4.63	4.81	56.39
$\mathbf{I}_{b,3}(F)$	-2.51	8.67	9.03	67.82	-2.11	3.59	4.16	67.15
$\mathbf{I}_{b,3}(L)$	-2.45	10.51	10.80	63.89	-2.13	3.64	4.21	63.38
$\mathbf{I}_{b,3}(P)$	-2.44	10.18	10.47	52.87	-2.10	3.77	4.32	52.49
$\mathbf{I}_{b,4}(F)$	-2.60	11.66	11.94	67.92	-2.44	3.43	4.21	67.48
$\mathbf{I}_{b,4}(L)$	-2.57	11.83	12.10	64.90	-2.39	3.33	4.10	64.50
$\mathbf{I}_{b,4}(P)$	-2.31	12.50	12.71	58.02	-2.16	3.56	4.17	57.66
$\mathbf{I}_{b,5}(F)$	-2.15	11.37	11.57	66.53	-1.99	3.51	4.04	66.12
$\mathbf{I}_{b,5}(L)$	-2.16	9.01	9.26	64.60	-1.90	3.31	3.81	64.15
$\mathbf{I}_{b,5}(P)$	-2.28	11.41	11.64	60.44	-2.09	3.51	4.09	60.06
$\mathbf{I}_{b,6}(F)$	-2.19	8.16	8.45	60.52	-1.92	3.23	3.76	60.08
$\mathbf{I}_{b,6}(L)$	-2.08	8.37	8.62	59.21	-1.84	3.13	3.63	58.80
$\mathbf{I}_{b,6}(P)$	-1.71	8.55	8.72	59.32	-1.41	3.26	3.56	58.88
$\mathbf{I}_{b,7}(F)$	-1.42	10.51	10.61	56.98	-1.29	3.42	3.65	56.74
$\mathbf{I}_{b,7}(L)$	-1.47	11.34	11.44	56.47	-1.34	3.31	3.57	56.21
$\mathbf{I}_{b,7}(P)$	-1.66	8.77	8.93	43.56	-1.38	3.39	3.66	43.28
$\mathbf{I}_{b,8}(F)$	-2.42	8.84	9.16	45.09	-1.99	3.72	4.22	44.78
$\mathbf{I}_{b,8}(L)$	-2.38	9.07	9.38	47.14	-1.98	3.56	4.07	46.86
$\mathbf{I}_{b,8}(P)$	-2.29	11.51	11.74	43.83	-1.73	4.64	4.95	43.55

Table 4.1: Results for multi-baseline matching, all values expressed in gsd[pix] except of $\%_{val}$ and $\%_{val,3}$. Each base images was matched with 4 neighbors using 3 types of rectifications. Rectification algorithms are [Fusiello et al., 2000] (F), [Loop and Zhang, 1999] (L) and [Pollefeys et al., 1999] (P).

Influence of Rectification Algorithms

Within a first evaluation the influence of the rectification types is evaluated. Therefore we treat the seven center images of the fountain data set as base image, each is rectified and matched against its two left and two right neighbors. For rectification three different methods as discussed in section 4.1 are utilized. Finally the depth maps D_r are generated using the multi-baseline approach described in section 4.3. The threshold for minimal consistent models were set to $t_{min} = 2$. The resulting statistics are depicted in table 6.5. As can be clearly seen there exists a systematic shift between the matched data $D_r(x,y)$ and the ground truth. This might be due to imprecise referencing of the image bundle to the LiDAR data or rest systematics in the camera calibration. For this reason a statement concerning absolute accuracy is not feasible. However, focusing on standard deviations and density of results tendencies due to the different rectification methods can be analyzed. As depicted in figure 4.18 Fusiello's rectification performs best regarding the average depth density. Loop's rectification delivers slightly reduced completeness but still is superior to results of polar rectification. Also, when evaluating σ_3, the standard deviation from which outliers $> 3\sigma$ were removed, the rectification types based on homographies yield best results. Here Loop's rectification yields slightly

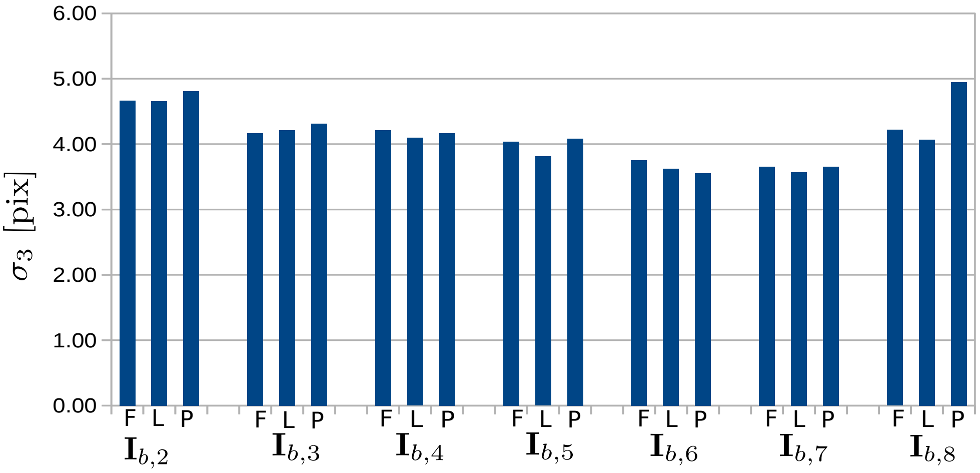
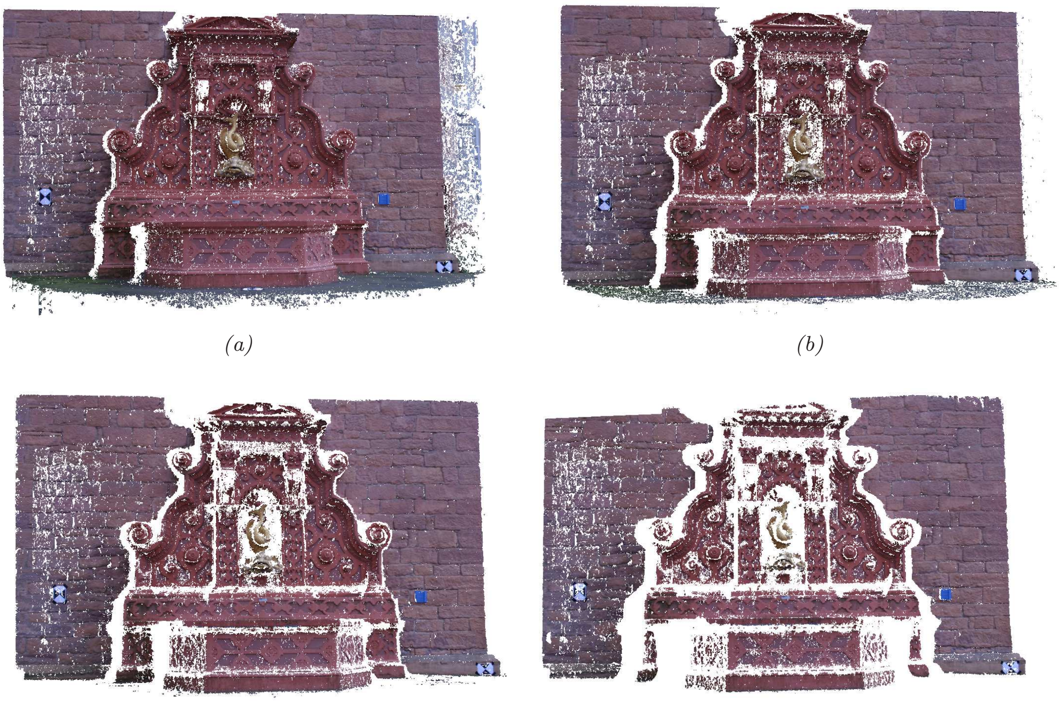


Figure 4.16: Standard deviation of 3σ fitered depth differences of 7 base images $\mathbf{I}_{b,2-8}$. Each image was matched against it 4 neighbors. Loop's rectification (L) in average yields best results followed by Fusioello's algorithm (F) and polar rectification (P).

better precision than Fusiello, because of the advanced mechanisms to reduce projective distortions, however differences are small compared to the absolute values. At this point it has to be mentioned that methods of forward intersection differ for homography based and polar rectification. Whereas within the latter object space errors are minimized for homography based methods image space errors are optimized. Therefore lower values for σ_3 may not only be due to the rectification type but also due to different triangulation methods. Slightly reduced densities at increased σ_3 for images $\mathbf{I}_{b,2}$ and $\mathbf{I}_{b,8}$ is due to larger variances in viewing directions and possibly less accurate orientations at image block borders. Although delivering worst results within this test, when dealing with pure forward motion polar rectification is the only technique which can be employed. One has to mention that for these scenarios ray intersection angles are typically small and depth precision is limited in particular for pixels located in the center of the images. Still, such observations can be used in combination with observations from wide-baseline stereo pairs for geometric consistency checks.

Correspondence Linking and Multi-View Forward Intersection

Within this evaluation the performance of the implemented depth map generation, based on correspondence linking, geometric consistency check and multi-view forward intersection is evaluated. As in the test before the fountain data set is utilized and the 5 central base images are rectified using Fusiello's method an subsequently matched against their 6 neighbors. Then the threshold of the number of minimal geometric consistent t_{min} is altered from $t_{min} = 1$ to $t_{min} = 5$. Depth maps D_r are compared to the ground truth depth maps using same statistical values as before. Again, a shift up to 3.3GSD is present which hinders a meaningful evaluation of absolute accuracies. However, tendencies of the algorithm behavior can be evaluated employing standard deviations and the percentage of successfully reconstructed pixels. Figure 4.19 depicts standard deviations of 3σ-filtered differences between ground truth and reconstructed depths. As can be observed σ_3 is continually decreasing as t_{min} is increased. This is a result of filtering outliers which posses low reliability due to reduced observations and more precise depths due to increased number of observations employed for multi-view forward intersection. The largest gradient can be observed in the transition from $t_{min} = 1$ to $t_{min} = 2$. For this transition all depths only observed by a single stereo model and the biggest part of outliers are filtered. When further increasing t_{min} lower values of σ_3 can be obtained. This is not primary caused by filtering further outliers but mainly is due to the larger number of redundant observations per depth. As depicted in figure 4.20 the percentage of successfully reconstructed pixels is continually



(a) *(b)*

(c) *(d)*

Figure 4.17: Visualization of point clouds for the base image $\mathbf{I}_{b,5}$. (a): $t_{min} = 1$ (M1), (b): $t_{min} = 2$ (M2), (c): $t_{min} = 3$ (M3), (d): $t_{min} = 4$ (M4)

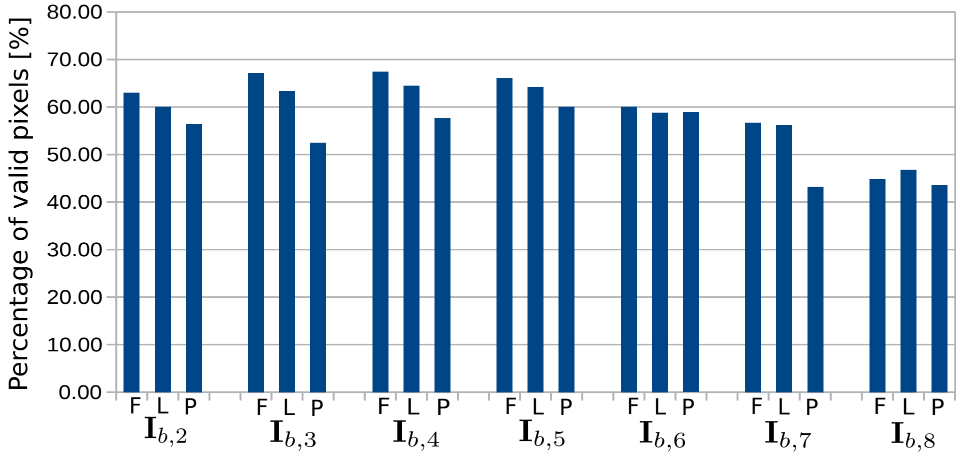


Figure 4.18: Percentage of reconstructed depths (3σ filtered) for 7 base images $\mathbf{I}_{b,2-8}$ with respect to the number of all image pixels. Each image was matched against it 4 neighbors. Fusiello's rectification (F) in average yields best results followed by Loop's (L) algorithm and polar rectification(P).

decreasing. Reduced number of reconstructed depths in the transition $t_{min} = 1$ to $t_{min} = 2$ is partly to outliers and depths only observed in single stereo models. For $t_{min} = 3$ a small amount of remaining outliers are removed but mainly depths are filtered due to limited observations in the images itself. Latter holds particularity for $t_{min} > 3$. Exemplary point clouds of $\mathbf{I}_{b,5}$ with $t_{min} = 1, ...5$ are depicted in figure 4.17. In practice for most cases we set $t_{min} = 2$ in order to obtain virtually outlier-free results at a good density, however at reduced precision.

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,3}(M1)$	-3.31	39.37	39.51	75.38	-3.00	10.88	11.28	74.54
$\mathbf{I}_{b,3}(M2)$	-2.31	12.50	12.71	65.16	-2.08	3.88	4.40	64.67
$\mathbf{I}_{b,3}(M3)$	-2.07	6.17	6.51	57.17	-1.77	2.72	3.25	56.52
$\mathbf{I}_{b,3}(M4)$	-1.71	4.13	4.47	45.95	-1.55	2.27	2.75	45.67
$\mathbf{I}_{b,3}(M5)$	-1.53	3.27	3.61	36.07	-1.41	2.02	2.46	35.84
$\mathbf{I}_{b,4}(M1)$	-1.60	44.92	44.95	73.69	-2.60	9.94	10.27	72.70
$\mathbf{I}_{b,4}(M2)$	-2.17	12.47	12.66	64.89	-2.04	3.44	4.00	64.49
$\mathbf{I}_{b,4}(M3)$	-1.94	6.85	7.12	58.14	-1.82	2.54	3.12	57.86
$\mathbf{I}_{b,4}(M4)$	-1.74	4.30	4.64	47.67	-1.64	2.16	2.71	47.42
$\mathbf{I}_{b,4}(M5)$	-1.48	3.14	3.47	36.23	-1.40	1.90	2.36	36.02
$\mathbf{I}_{b,5}(M1)$	-0.88	36.77	36.78	70.81	-2.60	9.12	9.48	69.94
$\mathbf{I}_{b,5}(M2)$	-2.12	12.73	12.91	62.11	-2.04	3.54	4.08	61.76
$\mathbf{I}_{b,5}(M3)$	-1.90	6.85	7.11	55.84	-1.83	2.51	3.11	55.59
$\mathbf{I}_{b,5}(M4)$	-1.73	3.88	4.25	44.36	-1.66	2.12	2.69	44.11
$\mathbf{I}_{b,5}(M5)$	-1.46	2.59	2.98	31.19	-1.37	1.74	2.22	30.85
$\mathbf{I}_{b,6}(M1)$	-1.44	37.12	37.15	62.76	-2.83	9.28	9.70	61.95
$\mathbf{I}_{b,6}(M2)$	-2.20	9.92	10.16	53.20	-1.97	3.44	3.97	52.83
$\mathbf{I}_{b,6}(M3)$	-1.88	5.55	5.86	46.47	-1.77	2.42	2.99	46.19
$\mathbf{I}_{b,6}(M4)$	-1.71	3.49	3.88	35.08	-1.61	2.03	2.59	34.82
$\mathbf{I}_{b,6}(M5)$	-1.97	2.92	3.52	13.51	-1.84	1.89	2.64	13.37
$\mathbf{I}_{b,7}(M1)$	-1.56	32.87	32.91	64.75	-2.11	9.79	10.02	64.14
$\mathbf{I}_{b,7}(M2)$	-1.52	11.24	11.34	53.39	-1.43	3.49	3.77	53.15
$\mathbf{I}_{b,7}(M3)$	-1.43	5.51	5.70	45.33	-1.34	2.47	2.80	45.12
$\mathbf{I}_{b,7}(M4)$	-1.85	3.15	3.65	28.32	-1.72	2.08	2.70	28.06
$\mathbf{I}_{b,7}(M5)$	-2.04	2.44	3.18	14.80	-1.91	1.59	2.48	14.60

Table 4.2: Results for multi-baseline matching, all values expressed in gsd[pix] except of $\%_{val}$ and $\%_{val,3}$. Each base images was matched with 6 neighbors using different thresholds of minimal geometric consistent observations $t_{min} = [1...5]$ (M1-M5). Image space accuracy was set to $\sigma_I = 2$.

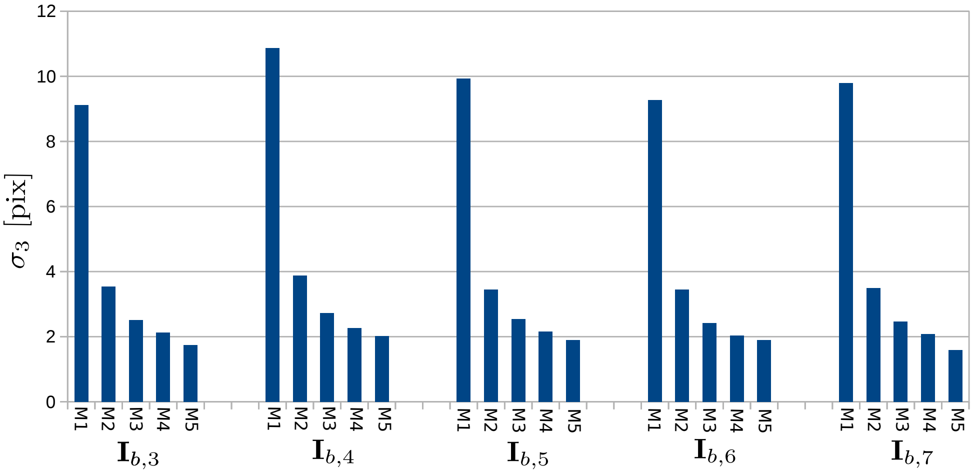


Figure 4.19: Precision of matched pixels for the baseimages $I_{b,3-7}$ in dependence of t_{min}, the number of minimal consistent disparities across the images (M1-M5).

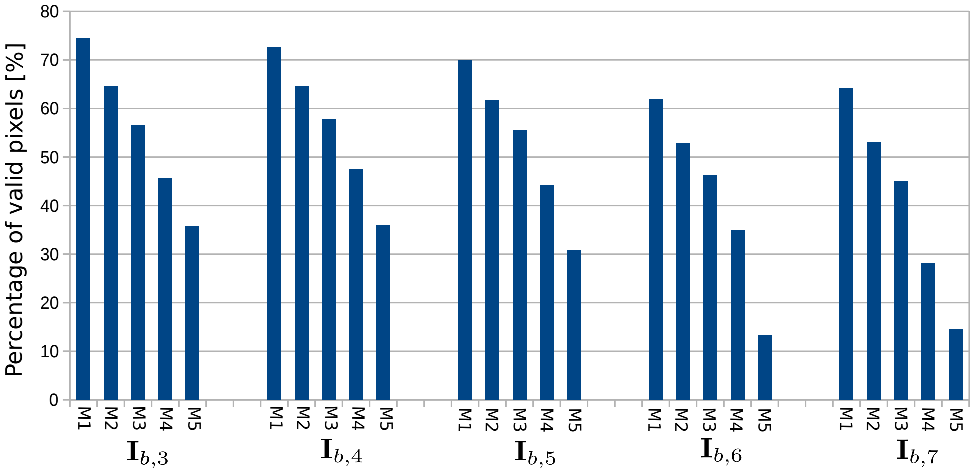


Figure 4.20: Percantage of matched pixels for the baseimages $I_{b,3-7}$ in dependence of t_{min}, the number of minimal consistent disparities across the images (M1-M5).

Chapter 5

Disparity Map Fusion for 2.5D Model Generation

In this chapter the fusion of depth maps to produce 2.5D elevation maps $e(x, y)$ is discussed. In the photogrammeric community two types of elevation maps are commonly used. Digital terrain models (DTM) or elevation models (DEM) are grids representing the actual earth surface. In contrast digital, surface models (DSM) map the earth surface including all objects (man-made structures, vegetation, etc.). Since image based methods reconstruct all visible objects, resulting 3D information is in particular suitable for the generation of DSMs. In the following chapter we use the terms elevation maps and digital surface model synonymously, DTM/DEM generation is out of the scope of this work. Within the fusion approach pixel-wise depth estimates as generated by the multi-view forward intersection (proposed in section 4.3) are fused to obtain a grid quasi-parallel to the reconstructed surface holding one elevation value per grid cell. Final cell-wise elevations are computed using median filtering with the filter direction defined by the normal of the terrain surface (approximated by a plane). After a detailed discussion of filter and interpolation strategies this chapter closes with an evaluation of the proposed algorithms. Thereby results are compared to those produced by other reconstruction algorithms from academia and industry. Elevation data can serve as a basis for the derivation of true-orthos, city models, cartographic products, etc. Another application is the visualization of surface models as for example for virtual globes. Because visualization most often is based on triangle meshes, it is desirable to transform the elevation data into a triangulated surface mesh. In section 5.4 we present our approach, which is based on a restricted quad-tree (RQT) adapting triangle sizes to local geometry. Moreover, a strategy for re-meshing facade triangles is presented, leading to improved robustness of normals which is essential for subsequent processing steps as texture mapping.

5.1 Fusion Strategy for 2.5D Elevation Models

For the generation of of 2.5D elevation maps we use a simple but effective gridding approach. Thereby a grid is defined quasi-parallel to the terrain surface. The points generated by the multi-baseline approach are then assigned to the grid, each point defining a height value of the cell in which it is located. Theoretically a point-wise precision according to equation 4.37 can be derived. However, these precisions not purely express geometric properties of the ray intersections, but also contain uncertainties of orientation and matching errors (fronto-parallel effects, blurred image regions, non-Lambertian surfaces, subpixel-locking, etc.), which are difficult to model. Therefore, the expressiveness of point-wise precision is rather limited. Moreover, outliers can not be detected based on precision values. For that reason we apply median filtering of the elevations contained in the final grid cells. Generally the size of a single grid cell is given by the average pixel footprint, however for low redundant data sets cell sizes are increased. The reason is that robustness with respect to outliers is only guaranteed with a sufficient number of elevations assigned to the single grid cells. Moreover,

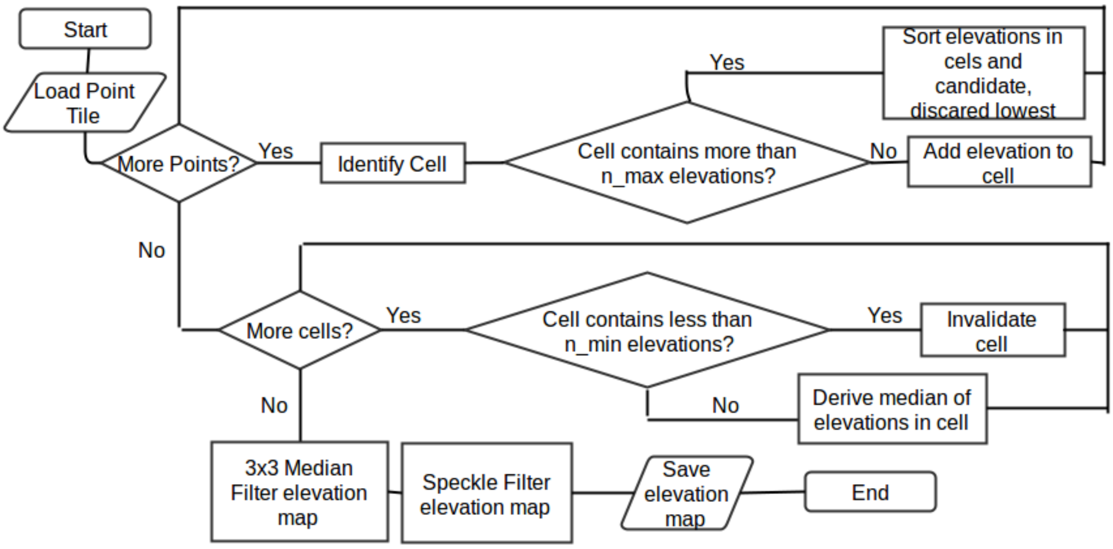

Start
Load Point Tile
More Points?
Yes
No
Identify Cell
Cell contains more than n_max elevations?
Yes
No
Sort elevations in cels and candidate, discared lowest
Add elevation to cell
More cells?
Yes
No
Cell contains less than n_min elevations?
Yes
No
Invalidate cell
Derive median of elevations in cell
3x3 Median Filter elevation map
Speckle Filter elevation map
Save elevation map
End

Figure 5.1: Flowchart of the algorithm for the generation of elevation maps.

we restrict the maximum number of elements contained by a single grid cell to n_{max}. The reason for this limitation is twofold. First, for high redundant datasets the assignment of each point of the generated depth maps would induce tremendous memory requirements. Second, particularly for data sets captured by wide angle camera systems, a considerable number of points representing 3D structure, as for example façades points, are extracted. These points hinder the generation of elevation maps since only 2.5D surfaces can be represented and generally the highest elevations in each grid cell are of importance. Not filtering the points representing 3D structure can lead to scattered building edges as visualized in figure 5.4 (left). In the example area marked by the green rectangle facade points are extracted below the roof. This causes median filtered elevations to alter between facade and roof points. Within our filter approach we set n_{max} to the number of average detections per grid cell. This value can be derived from the initialization process. If a candidate point is added to a grid cell which already contains n_{max} elevations, existing elevations and the candidate are sorted and the lowest elevation is discarded. After all points have been assigned to the grid, we employ cell- wise median filtering which defines the final elevation. Figure 5.4 (right) shows the result eliminating the scattering effects by limiting the maximal elements per cell and therefore favoring the highest (roof) points. To guarantee robust results we only consider cells for which at least n_{min} candidates were assigned. Due to the moderate number of outliers produced by the multi-baseline approach $n_{min} = 3$ is typically sufficient. The final elevation maps are then subject to speckle filtering and median filtering using a kernel size of 3×3. The algorithm for the generation of elevation maps is summarized in figure 5.1. Limiting elevation buffers to n_{max} reduces the memory footprint, however, large scale projects require a tiled processing scheme. Thereby points generated by the multi-baseline triangulation are not stored image-wise but tile-wise. Practically points are appended to a file containing all points within a bounding box defining a certain spatial extend. The proposed fusion strategy for elevation maps is then carried out for each single tile. This way we guarantee to load and process each point only once to avoid additional input/output as well as processing overhead. Since the filtering is applied in the direction orthogonal to the surface filtering artifacts at tile borders are prevented.

5.2 Interpolation of Elevation Maps

Due to limited redundancy in data sets, occlusions and sparse dense matching results in areas possessing challenging texture, not all of the grid cells contain the critical amount of n_{min} elevation candidates. For

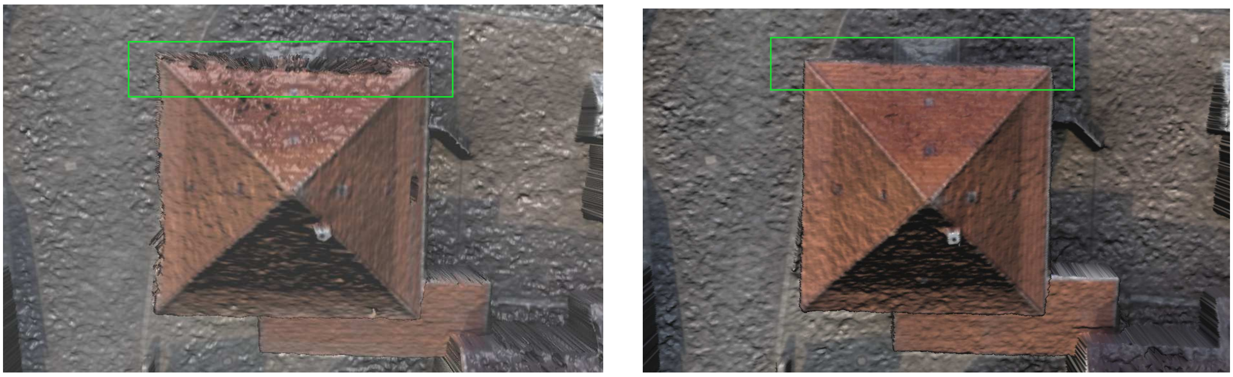


Figure 5.2: Left: Visualization of meshed DSM derived by median. Right: Visualization of meshed DSM derived by the proposed approach where 3D geometry (facade structure) is filtered which improves reconstruction of clean roof edges.

these cells elevation values are derived by interpolation. Typical interpolation methods for DSM data are based on inverse weighted distance(IWD) or natural neighbors. Due to beneficial processing times our method is an adapted version of the inverse weighted distance interpolation

$$e = \frac{\sum_{i=0}^{N} \frac{e_i}{d_i}}{\sum_{i=0}^{N} \frac{1}{d_i}}. \tag{5.1}$$

Thereby e_i defines an elevation value and d_i is its distance to the pixel to be interpolated. The elements e_i are derived by scanning the four quadrants beside the invalid pixel to be interpolated for valid elevations. The nearest valid elevations are incorporated into the interpolation process. However, especially in build-up areas this interpolation strategy leads to blurred building edges. These effects occur in particularly for low redundant flight configurations for which large occlusions are present. Inaccurate building edges cause artifacts in subsequent processing steps as for example in the course of true-ortho production since this process heavily relies on surface geometry. The second interpolation method is designed to tackle this problem. It is based on the assumption that cells for which elevations are not available are occluded cells and the probability that they are part of low elevation areas is larger than that they belong to areas of high elevation. Therefore, these cells are interpolated from neighboring cells possessing low elevation values solely. For each pixel to be interpolated the closest valid elevations on 16 image paths intersecting the pixel are identified. The path directions are specified similar to the paths within the SGM-approach (vertical, horizontal, diagonal). Let e_{min} be the smallest elevation in the set of 16 elevations e_n, $n = [1, ..., 16]$ (figure 5.3). An elevation e_n is incorporated into the interpolation process if

$$e_n - e_{min} < t_e. \tag{5.2}$$

Thereby t_e is a positive threshold typically set 1.5 meters. Beside the elevations e_{min} and e_n also the respective distances d_{min} and d_n are recorded. The final elevation is derived by inverse weighted distance interpolation (following equation 5.1) using all elevations satisfying the criterion 5.2. To guarantee reliable results thresholds for the number of minimal detections in a certain search radius are applied. These restrictions can be applied per quadrant as well as for the whole search window around the pixel to be interpolated.

5.3 Results

To demonstrate the capabilities of the implemented pipeline the generated elevation data is compared to results of commercial software packages and state-of-the-art DSM extraction implementations developed by

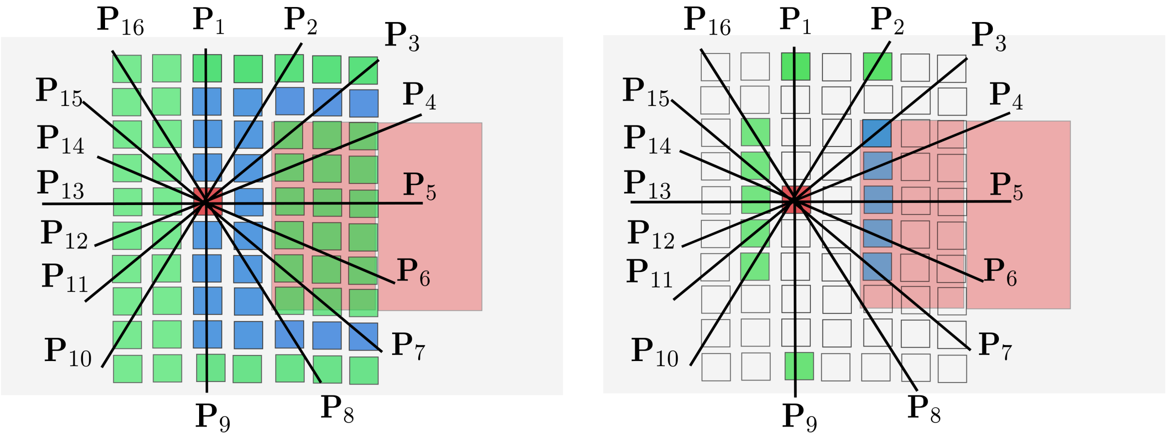


Figure 5.3: Visualiztion of the interpolation process. Gray rectangle depicts elevations possessing low elevations, red rectangle depicts area of high elevations. Left: Green squares mark valid elevations, blue square mark invalid elevations. Right: The nearest valid elevations along each of the 16 image paths are identified. Only low elevations (green squares) are utilized for inverse weighted distance interpolation and high elevations (blue squares) are excluded.

mapping agencies, research institutions and universities. Elevation models utilized for comparisons were produced by users and vendors of the respective software packages in the course of an EuroSDR initiative 'Dense Image Matching Benchmark' [Haala, 2013a]. Since evaluating the final elevation data, the comparison not only comprises dense multi-view matching but also 3D point triangulation, subsequent fusion algorithms and interpolation methods. Product or vendor names are not mentioned in this section because this is of minor importance to the evaluation. The gross of methods utilize variations of semi-global-optimization, however some of the matching algorithms are based on total variation, belief propagation and graph cuts. Some of the methods incorporate multi-photo consistency measures others are based on pure stereo matching. The test can not be considered as an in-depth evaluation of optimization strategies or dense matching algorithms since for most solutions details of implementations are not available and, as mentioned before, final elevation data is not dependent on the dense matching algorithm only. Resulting reconstructions are dependent also on fusion strategies, filtering and interpolation approaches. Rather, this evaluation should clarify how our pipeline compares to other DSM generation algorithms and what reconstruction results can generally be expected for airborne mapping applications. A similar study was conducted by [Haala, 2013b] which is based on the same DSM data. The test data comprises 15 large frame (14144 × 15552) DMC images along with image orientations computed using the commercial software package Match-AT. Overlaps of imagery amount 80% in flight direction and 80% between flight strips with a nominal ground sampling distance of 10cm. Our results were produced by matching each image with its two neighbors in strip and two neighbors cross strip and applying the algorithm for DSM extraction proposed in this chapter. We subdivide this section in two parts, first evaluating time demands by the single algorithms and then discussing quality of the generated elevation data.

5.3.1 Processing Times

Table 5.3.1 displays the processing times of the single algorithms. Because results were not produced using identical hardware, computation times are difficult to compare. However, for a rough evaluation we approximated the time t_s which the different solutions would need to generate the DSM on a single core 1GHz processor. This value was computed based on the number of physical CPU cores and clock speeds of the participants. Of course this is only a rough approximation neglecting many factors including available memory, memory bandwidths, caching capabilities, over-clocking modes/boosting, threading mechanism, etc. Moreover, some solutions employ GPUs and FPGAs. For these devises approximations were computed

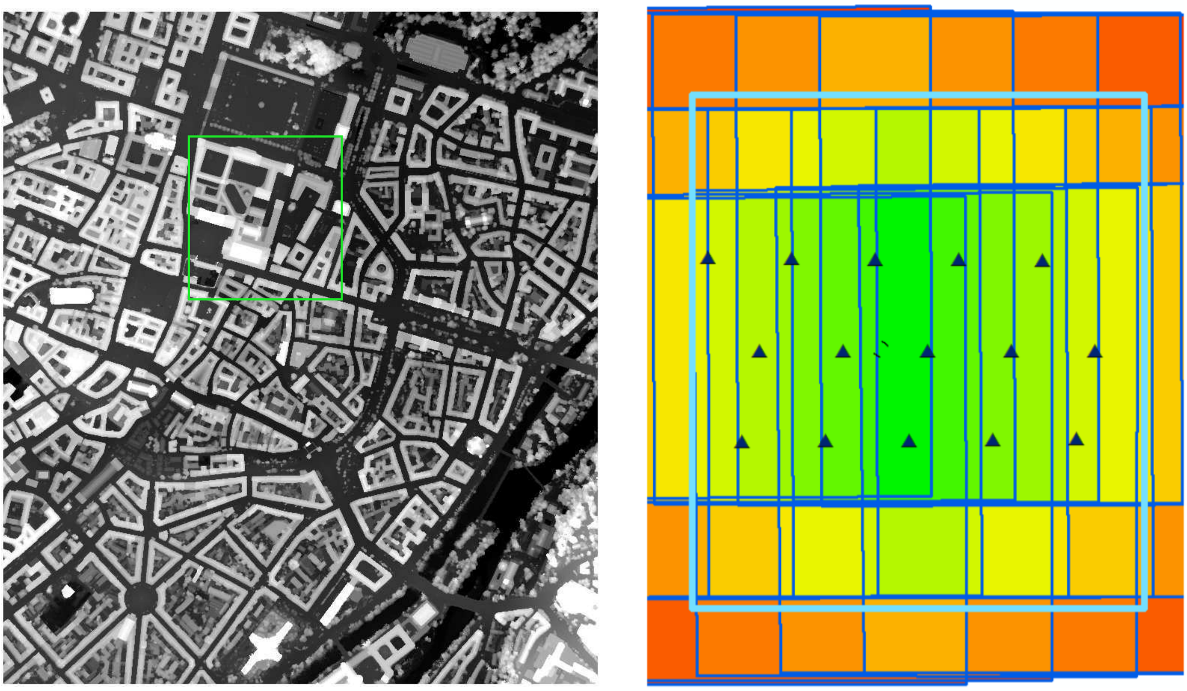


Figure 5.4: Left: Median DSM computed from all solutions. Green box depicts the area for which profile analyses and quantitative evaluations are carried out. Right: Visualization of the image block comprising 25 DMC2 images (80% forward / 80% side overlap) taken from [Haala, 2013b]. Cyan box depicts the area for which median DSM was calculated.

based on utilized CPUs only neglecting computation capabilities of the additional devices (these numbers are marked by * in table 5.3.1).

Solution	CPU cores/clock[Ghz]	GPU	FPGA	time[h]	t_s[h]
A	4 / 2.93	no	no	25	293
B	6 / 2.3	5	no	0.45	6.21*
C	4 / 3.7	no	no	19	281.2
D	12 / 2.5	no	no	2,2	66
E	30 / 2.4	no	no	5	360
F	8 / 2.0	no	no	21.65	346.4
Ours	4 / 3.4	no	no	4.62	57.39
H	4 / 3.6	no	yes	5.85	80.35*

Table 5.1: Time comparison of the different algorithms. * mark hardware configurations using GPUs or FPGAs. t_s is the time scaled to a one core processor running at a clock speed of 1GHz.

The fastest method regarding total time is solution B utilizing 5 GPU devices along one CPU. This results in extremely fast processing times. A FPGA solution implementing the classical SGM extracted surfaces in $t_s = 80.35$. Note that this number is larger than some solutions which use weaker CPUs and no additional FPGA device. Maximal processing time was recorded by solution F, however, since distributing the processing over 30 nodes a total runtime of only 5 hours could be achieved. Our algorithm favorable compares to the other methods ranking place 3 with respect to total time used, and ranking first place regarding the scaled time t_s for solutions employing CPUs solely.

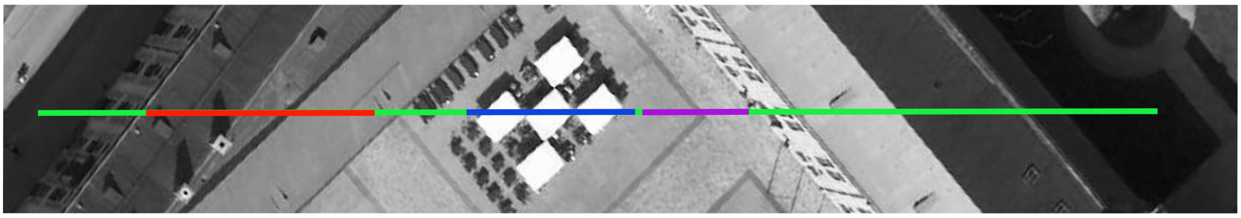


Figure 5.5: Pan chromatic image with visualization of slopes on which our profile analysis is based on. Green: complete slope. Red: roof slope. Blue: umbrella slope. Purple: ground slope.

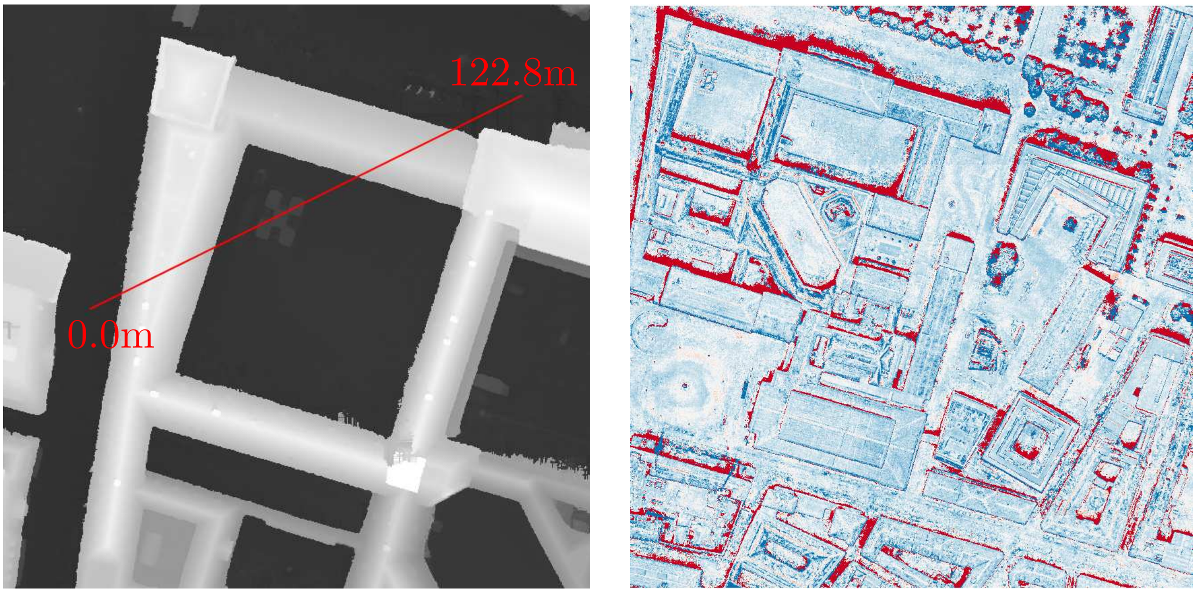


Figure 5.6: Left: Slope on which our profile analysis is based on. Right: Visualization of differences, the median map subtracted from our solution. Blue and red depict differences larger and smaller than 0.3m (3GSD).

5.3.2 Differences and Precision of Benchmark DSMs

Unfortunately ground truth is not available for the test area. To compare the differences of our algorithm and other solutions we compute a median map (see figure 5.4). More precisely we assign all elevations of all solutions to a grid similar to the DSM data and compute the median elevation per grid cell. At this point it has to be mentioned that elevations produced by algorithm (C) contained many outliers and was discarded for the following evaluation. Since no ground truth data is available no conclusions about absolute accuracies can be drawn. However, analyzing differences of the single solutions give a hint of achievable precision. Because full redundancy is only available in the center area of the test site these evaluations are performed on an rectangular subpart of the image depicted by the green box in figure 5.4. In figure 5.6(right) the difference of our DSM to the median DSM is visualized. Thereby differences are color coded in a range of ±3GSD. For numerical analysis we removed elevations larger than ±10GSD. These differences are mostly due to low redundant observations beside buildings as a result of occlusions. In these areas differences are mainly determined by the utilized interpolation methods. After computation of mean and standard deviations according to equation4.41, all elevations were filtered using the 3σ criterion. After filtering, the resultant mean difference amounts 0.09 meters which indicates a systematic shift of our solution below one GSD. The standard deviation amounts 0.27 meters in a range below three GSD.

Within a second more qualitative evaluation we analyze elevations and elevation differences along the

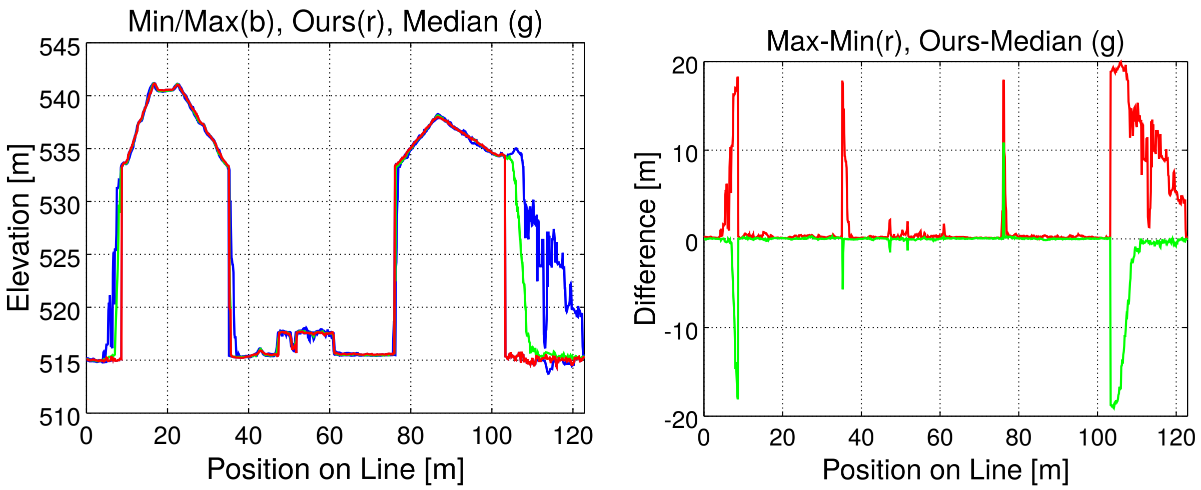


Figure 5.7: Profiles for the complete slope (figure 5.5, green). Left: Min/Max elevations (blue), Median elevation(green), Our elevation (red). Right: Difference of max/min elevation(red), Difference of our and median elevation (green).

slope displayed in figure 5.6(left). The length of the slope amounts 122.8 meteres, crosses two buildings and covers flat fronto-parallel surfaces, small structures with low radiometric information and areas possessing low number of observations due to shadows and occlusions. Four single profile sections are analysed and visualized in figure 5.5. We analyze the single profiles using minimum and maximum elevations generated by all solutions as well as the median profile and the profile of our solution. For better visualization we also plot the difference between our and the median surface, denoted by δ_{om} and the differences between minimum and maximum elevations δ_{mm}.

Complete Slope - Shadow and Occluded Areas Figure 5.7 depicts elevations along the complete slope. Blue lines correspond to minimal and maximal elevations of all submitted elevation grids, the green line displays the median elevation and the red line encodes heights extracted by our algorithm. As can be seen main differences are located in the occluded/shadowed area (100-110m). Most of the solutions extract noisy elevations and most interpolation methods produce blurred transitions at height discontinuities. However, the interpolation proposed in section 5.2 in combination with the robust geometric consistency and DSM filtering results in favorable reconstruction of clear building edges in elevation maps generated by our algorithm. This can be also observed for height discontinuities in non-occluded areas. Figure 5.7(right) highlights the difference of minimal and maximal elevations of all solutions depicted by the red line. Furthermore, the green line encodes the difference between the median elevations and elevations reconstructed by our algorithm. This color coding is maintained in the course of all further profile analyses. Similar to the discussion before sharp peaks occur at height discontinuities. Narrow peaks are due to differences in the reconstruction algorithms not handling abrupt height discontinuities well, the wide peak at the end of the slope (100-110m) is mainly due to different interpolation methods. In the section 110-120m high variances are caused by unbeneficial radiometry due to shadows.

Roof Slope - Fronto Parallel Effects / Object Borders Figure 5.8 shows profiles of a roof section. Color coding is identical to the chart discussed before. This part of the slope contains mostly non-fronto parallel surfaces except in the range 18m-21m. Figure 5.8(right) depicts increased residuals for minimal and maximal elevations for surfaces not parallel to the image plane. In other words, variances of the single solutions are larger. These fronto parallel effects are due to modeling errors, for example window based similarity measures and smoothness assumptions forcing extracted surface being parallel to the image plane.

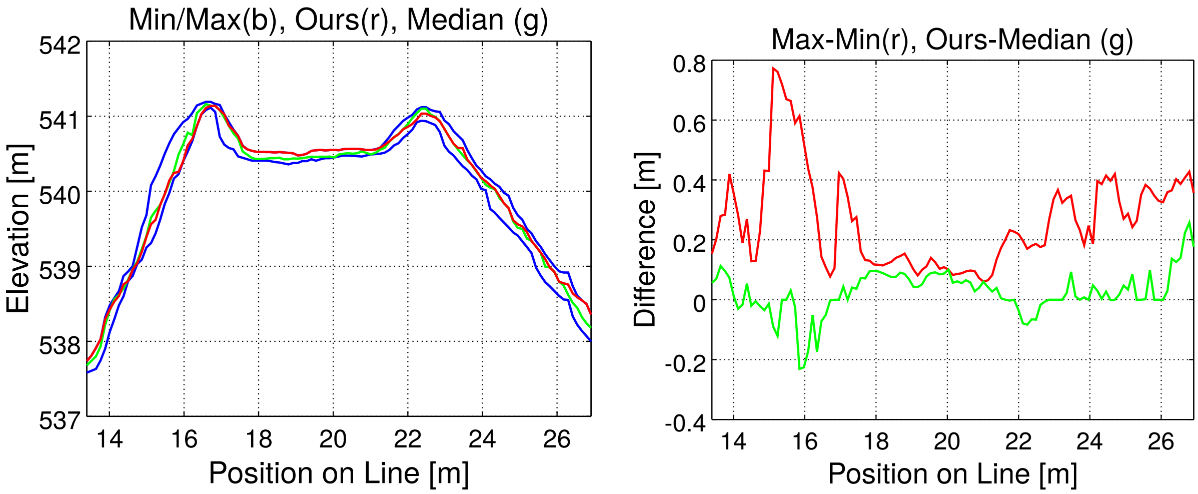


Figure 5.8: Profiles for the roof slope (figure 5.5, red). Left: Min/Max elevations (blue), Median elevation(green), Our elevation (red). Right: Difference of max/min elevation(red), Difference of our and median elevation (green).

Furthermore, this is partly due to the evaluation procedure itself: residuals would decrease if computation of residuals would be performed in direction of the actual surface normals. In these areas the different solutions vary largely and yield a δ_{mm} of up to 8 GSD. For the fronto parallel slope part our solution differs from the median by 1 GSD. Maximal residuals for slanted surfaces are below 3 GSD.

Umbrella Slope - Minimal Radiaometric Information The third slope is depicted in figure 5.5. The scene contains white umbrellas which are over exposed in the single images. Therefore reduced radiometric information is available and rather low signal-to-noise ratios can be expected. As can be observed in figure 5.9 our algorithm performs favorably at object boundaries. For these part also largest residuals δ_{mm} and δ_{om} are observed. The umbrellas themselfs are reconstructed possessing an acceptable noise level by our algorithm. Whereas σ_{mm} yields differences in a range of up to 9 GSD our solution possess residuals with respect to the median map of below 2 GSD.

Ground Slope - Low/High Radiometric Information The fourth slope section depicts a fronto-parallel surface as can be seen in figure 5.5, in the beginning of this slope section radiometric information is rather limited, in second half more texture is available as a result of pebble on the ground. As can been seen in figure 5.10 for the fist section ranges of minimal and maximal elevation differences amount 2.5 GSD. Variances decrease for areas possessing beneficial texture information. For these parts a σ_{mm} close to 1.5 GSD is obtained. Our solution seems to be more robust in low textured areas possessing differences to the median elevation in the range of 1 GSD.

5.4 Meshing of 2.5D Elevation Maps

For airborne nadir scenarios extraction of 3D structure is challenging due to restricted viewing directions and occlusions. Typically the combination of limited number of observations and disposedness of surfaces results in only sparse reconstructions of 3D geometry, for example at façades. Reconstruction of geometry presumes at least two redundant observations in the images. More observations are required if additional restrictions regarding geometric consistency across stereo models are claimed. Therefore, the single images contain texture information of areas for which no geometric information is available. This holds particularly true for building facades as these are often occluded in a large number of views. However, the radiometric information in the single images can be utilized for surface texturing. Due to the lack of real 3D geometric information

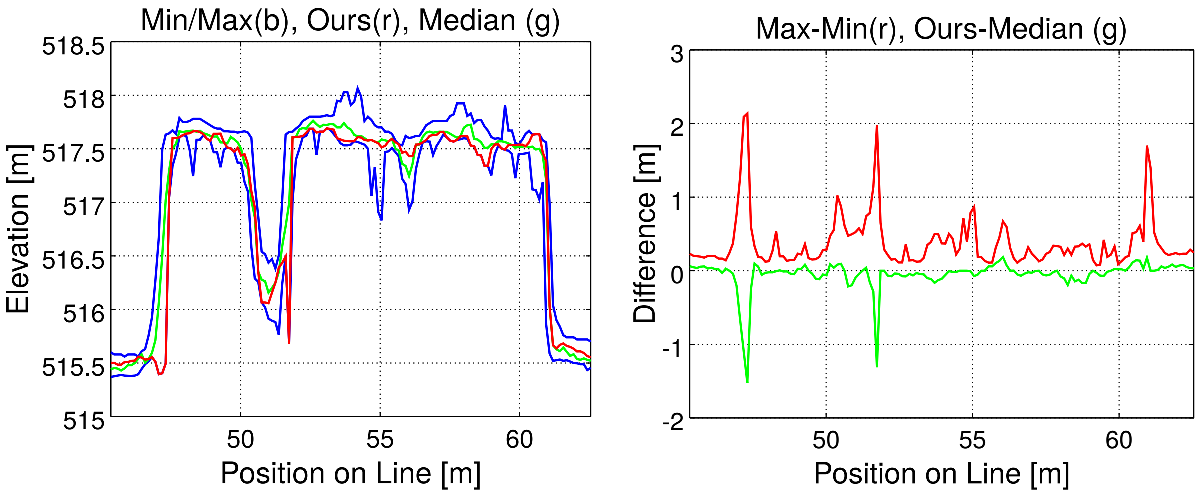


Figure 5.9: Profiles for the umbrella slope (figure 5.5, blue). Left: Min/Max elevations (blue), Median elevation(green), Our elevation (red). Right: Difference of max/min elevation(red), Difference of our and median elevation (green).

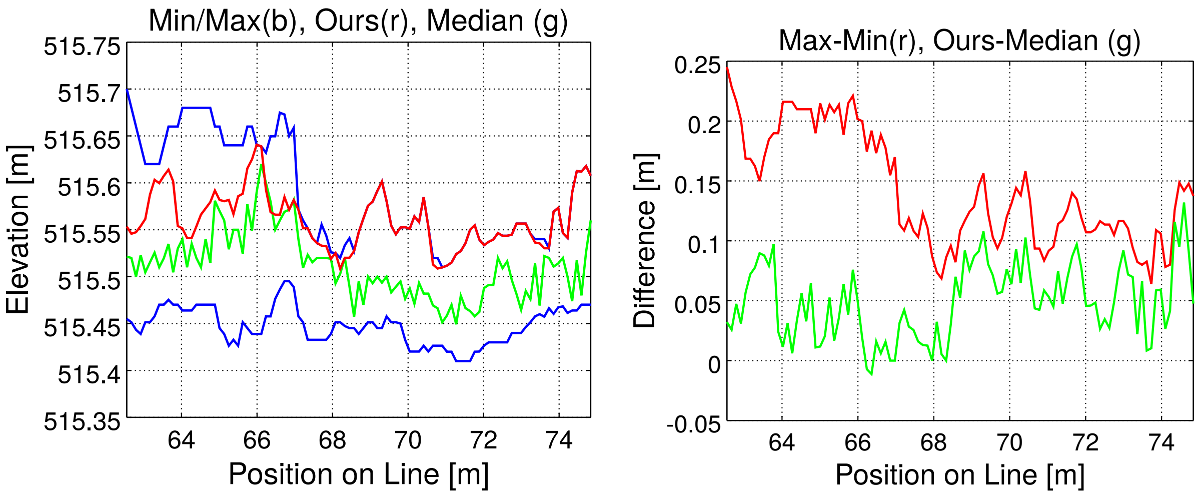


Figure 5.10: Profiles for the ground slope (figure 5.5, purple). Left: Min/Max elevations (blue), Median elevation(green), Our elevation (red). Right: Difference of max/min elevation(red), Difference of our and median elevation (green).

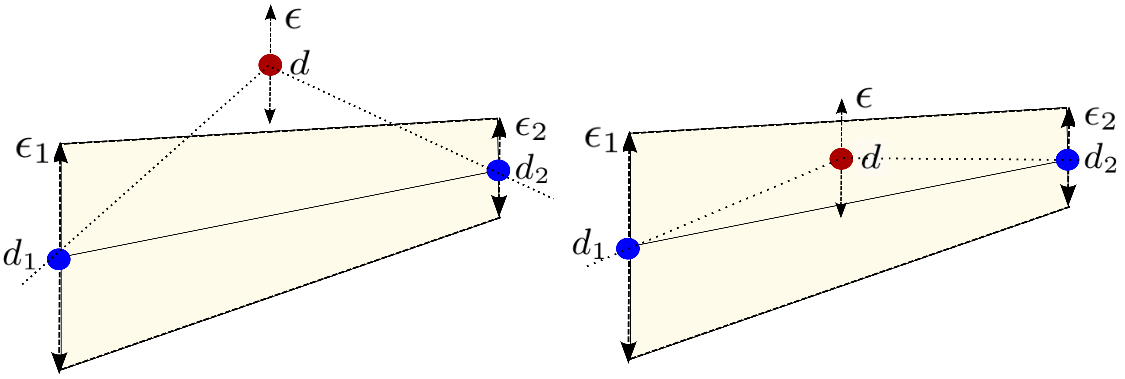


Figure 5.11: Visualization of the error criterion, elevation candidate depectied in red, two elevations already added to the mesh depicted in blue. Left: error criteria is not fulfilled, elevation candidate is inserted to triangulation. Right: error criteria is fulfilled, point candidate is not inserted to triangulation.

we generate meshes based on the 2.5D elevation data and subsequently apply texture mapping. Complexity of visualization and texture mapping as well as other post processing tasks depend on the number and density of triangles. Therefore, it is desirable to construct possibly light-weight meshes. On the other hand, if elevation meshes are simplified to a high degree, geometric details of the surface are removed and texture mapping will deliver poor results. As a consequence we seek to only consider elevation data contributing to the actual geometry and neglect all other. An ideal framework to tackle this problem is given by restricted quad-trees (RQT) [Pajarola, 1998]. Since operating on 2D space the triangulation process is computational efficient compared to 3D approaches. However, in areas of large depth variations, in particular at facades, the algorithm produces triangles possessing large triangle side ratios possessing unreliable normals. This directly affects subsequent processing steps which rely on robust normals. Therefore, we re-mesh the problematic areas as will be explained in section 5.4.2.

5.4.1 Meshing of Elevation Data Using Restricted Quad Trees

As mentioned before, we utilize restricted quad trees (RQT), a subclass of the quadtree, for the generation of meshes from elevation data. Region quadtrees are data structures spatially separating 2D space. Starting from an initial quad (node), each quad is recursively split into four sub-quads (sub-nodes) until a certain level is reached or each node satisfies a specific criteria. Quadtree triangulations of 2.5D elevation data were covered in multiple works, mainly in the field of computer graphics, for regular and irregular data points, see [Sivan and Samet, 1992], [Pajarola, 2002], [Pajarola et al., 2002] for an excellent overview. In [Pajarola, 1998] a special type of this structure, a restricted quadtree, was used for the triangulation of digital elevation data for the purpose of terrain visualization. The idea is to build a RQT on regular 2.5D height data raster from which then a simplified triangulation can be derived. One of the key properties of RQTs is the option to extract matching triangulations from regular grids, meaning that resultant surfaces represented as triangle meshes are crack-free. Moreover, all vertices of such triangulations are guaranteed to fulfill specific error criteria. We exploit latter property to discard elevation estimations not contributing to the geometry at an early stage.

Generally there are two different approaches for the RQT construction. The bottom-up strategy initially starts by defining nodes on the lowest (full resolution) level. Then, each node is evaluated based on a quadtree and an error criterion. The quadtree criterion assures that side lengths of proximate grid cells are identical, the half or the double. This property allows to obtain matching triangulations. The error condition, typically the approximation error, describes to what extend geometric errors are introduced if a particular node is removed. If these two conditions are met, nodes are fused and higher level nodes are

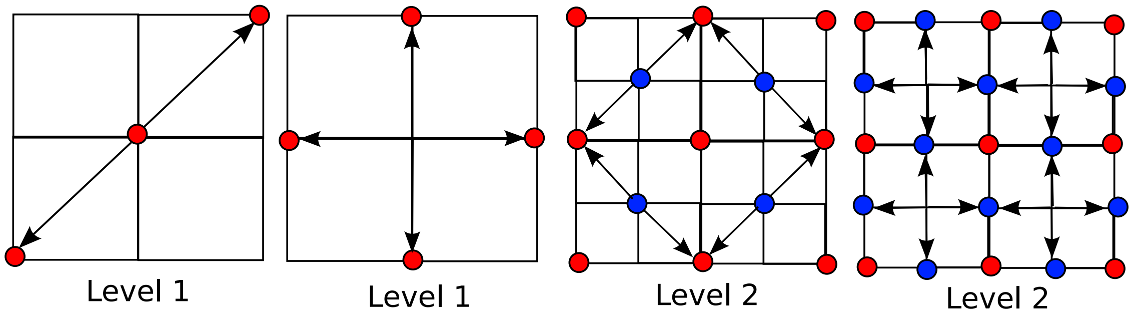


Figure 5.12: Visualization of two subsequent levels of the dependency graph structure. Red nodes depict level 1 nodes (elevations), blue nodes depict level 2 nodes(elevations)

examined. The major draw back of this approach is that error criteria can not be guaranteed. Although these conditions are fulfilled for nodes of lower levels the errors accumulate over multiple levels and are not guaranteed to be fulfilled on higher levels. This drawback can be overcome by a top-down construction strategy. Thereby initially the raster data is represented by a single node. Nodes are added recursively if error criteria are not fulfilled. However, compared to the bottom-up methods the implementation is more sophisticated because when adding a node on level l also updates on higher levels $l+n$ might be necessary to maintain the restricted quadtree structure, e.g assuring that the quadtree criterion is fulfilled. An efficient implementation is given in [Pajarola, 1998] which is based on dependency graphs. As visualized in figure 5.12 each node depends on two other nodes from the same or the next higher level. Recursively, in a top-down manner, all nodes are checked for error criteria based on their two dependent nodes and activated if these are fulfilled. If a node is evaluated to be a part of the triangulation then its dependent nodes and their dependencies have to be activated too. This can be conveniently implemented by recursive functions. From the set of finally activated nodes a matching triangulation can be extracted by skimming activated nodes bottom-up.

In the following we explain the error criteria used to construct meshes from 2.5D elevation data only considering elevation points contributing to the geometry. As mentioned before simplification is crucial, since the number of triangles directly affects further visualization performance and post-processing steps as texturing. We base our dynamic error criterion on the cell size of the elevation grid. Let n be a node with the elevation $d = e(x, y)$ and its dependent nodes n_1, n_2 with $d_1 = e(x_1, y_1)$ and $d_2 = e(x_2, y_2)$ respectively. Each elevation is assumed to be reconstructed with an uncertainty of $t \cdot gsd$. The node n is inserted to the triangulation if it is not contained in the local noise band which is defined by $d_1 \pm \sigma$ and $d_2 \pm \sigma$ (figure 5.11). Hence, a depth value is considered to contribute to the geometry if

$$d - \epsilon > \frac{d_1 + d_2 + \epsilon_1 + \epsilon_2}{2} \quad or \quad d + \epsilon > \frac{d_1 + d_2 - \epsilon_1 - \epsilon_2}{2}. \tag{5.3}$$

The memory requirements of data structures for dependency graphs limits the size of triangulations. However, our RQT framework in combination with the tiled processing scheme offers the possibility to mesh elevation data of arbitrary size. To generate matching triangulations at tile borders we load all neighboring elevation tiles corresponding to a center tile and add vertices at the border grid cells to the triangulation even if error criteria are met. In figure 5.15 (right) this process is visualized. Neighboring tiles are marked by red and gray background colors. The triangulation of the left tile is carried out on its elevations and elevations in the transition zone, in this example the first column of the right tile. If not already part of the triangulation, vertices are inserted at elevation cells q_{1-6} and p_{1-6}. The vertices in the transition zone as well as the faces connecting p_{1-6} and q_{1-6} are part of the mesh of left tile. These entities connect meshes from neighboring tiles. As a result the approach scales well to large datasets and enables the generation of matching triangulations for country-wide projects.

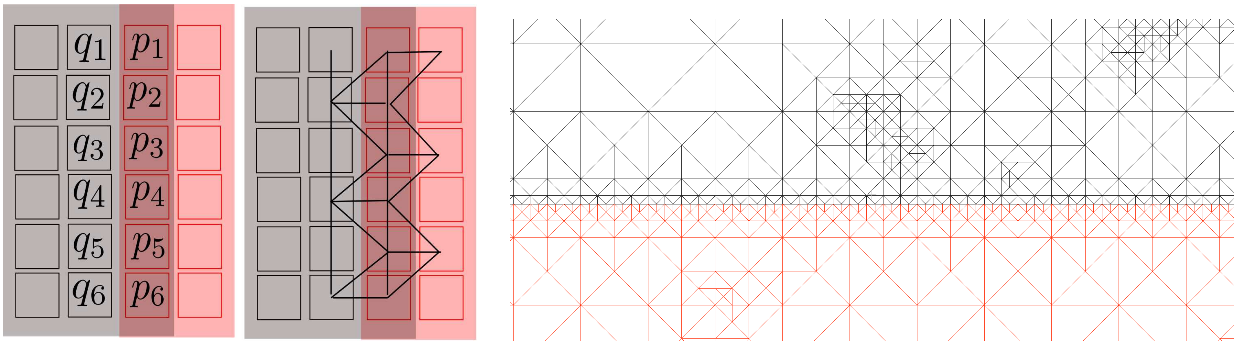


Figure 5.13: Left: Visualization of the approach guaranteeing that meshes of neighboring elevation tiles are seamless. Consistent borders of two neighboring mesh tiles (black and red) can be obtained by adding additional vertices close to the tile border. More details are given in the text. Right: Visualization of the mesh structure derived from elevation data using a RQT at tile borders.

Algorithm 2 Algorithm for re-meshing of facade triangles.

```
Require: Re-mesh facade triangles possessing bad large triangle side ratios
Input: Matching surface triangulation $\mathcal{S}$ with vertices, $\mathcal{V}$, triangles $\mathcal{T}$. Subdivision parameter $e_{max}$.
Output: Refined matching triangulation $\mathcal{S}'$
Initialization: Lock all vertices
Selection 1: Identify all vertices which are part of triangles possessing a bad triangle side ratio. This set is denoted by $\mathcal{V}'$. $\mathcal{T}'$ defines the subset of triangles possessing at least one $v \in \mathcal{V}'$.
Split: Split triangles $t \in \mathcal{T}'$ until all edges are smaller than $e_{max}$
Selection 2: Identify all vertices which are located at the tile borders. This set is denoted $\mathcal{V}'_b$.
Smooth 1: Smooth vertices $v \in \mathcal{V}' \backslash \mathcal{V}'_b$
Merge: Merge triangles $t \in \mathcal{T}'$ based on quadric error metrics and triangle side ratio.
Smooth 2: Smooth vertices $v \in \mathcal{V}' \backslash \mathcal{V}'_b$
```

5.4.2 Re-meshing of Facade Triangles

As mentioned before the RQT approach inserts triangles possessing large side ratios at depth discontinuities, for example at building facades. For these triangles unreliable normals can be expected due to planar errors of vertices corresponding to high and low elevation points which are propagated over the whole façade. To improve results in the texturing process and for visually appealing surface meshes we re-mesh these problematic areas. One key concern is that after the re-meshing procedure tile-wise triangulations are still matching, in other words no cracks are introduced at tile borders. The re-meshing process is carried out per tile meaning that no information from neighboring triangulations is required. In the course of re-meshing we identify badly shaped triangles by the triangles side ratio. These faces then are subdivided until the a minimal side length of each triangle equal or smaller than e_{max}. We set e_{max} dependent on the GSD, usually $e_{max} = 10 \cdot gsd$. The generated triangles are subject to a Laplacian smoothing in order to reduce noise. Since triangles are small, only minimal errors regarding geometry are introduced. In this step it has to be assured that newly introduced vertices located at the tile borders are not modified, therefore we discard them from the smoothing process. Within the next step triangles are fused again using quadric error metrics [Garland and Heckbert, 1997] to preserve as much geometry as possible. Note, that within the simplification process the vertices derived by the RQT are locked, meaning they are not considered in the fusion process. This way the actual geometry extracted by the RQT is not changed. As a result error criteria remain valid and triangulations of neighboring tiles remain crack-free. Finally, the fused facade triangles are subject to a Laplacian smoothing to increase robustness of normals. As before vertices located at tile borders are not considered. The algorithm for facade re-meshing is summarized in the algorithmic chart

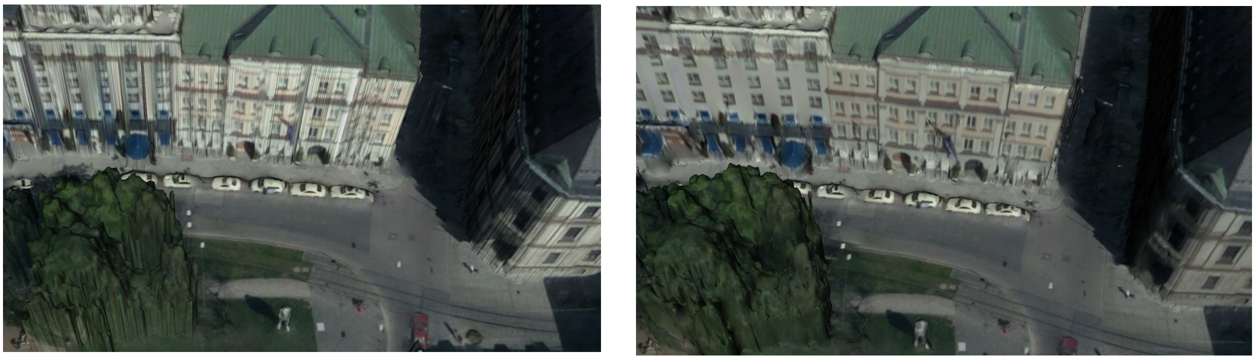


Figure 5.14: Left: Textured mesh without facade re-meshing. Right: Textured mesh with re-meshing.

2. The implementation of smoothing, subdivision and triangle fusion are based on the OpenMesh library [Botsch et al., 2002].

For visualization purposes these triangulations can now be textured from the available imagery. In general a triangle of the mesh is seen in multiple views which arises the question what image should be used to texture the face. Using single views for texturing instead of blending avoids blurred textures and ghosting effects in areas where the scene is not static (for example moving cars). To derive consistent colors we follow the approach proposed by [Waechter et al., 2014] which cast the task as an energy optimization problem selecting the most adequate image for texturing a face whilst forcing neighboring faces to be colored from the same image. An example of a textured surface mesh generated by the presented approach using nadir imagery only is displayed in figure 5.15 (lowest), even better results can be obtained by using additional oblique views. Figure 5.14 depicts the difference of mesh texturing with and without the re-meshing procedure.

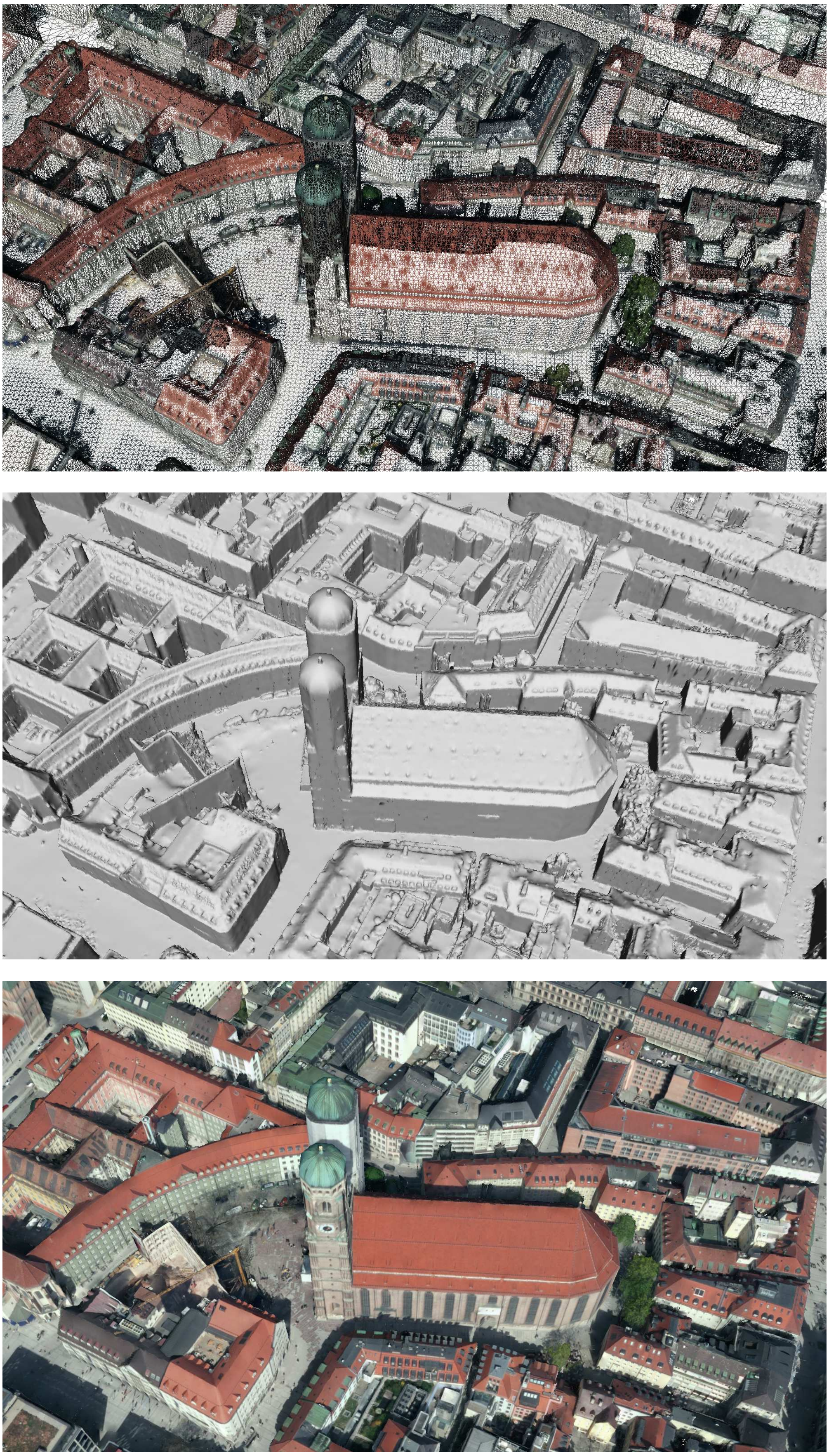


Figure 5.15: Top: Mesh generated by the RQT approach including facade re-meshing. Middle: Shaded visualization. Bottom: Textured mesh.

Chapter 6

Disparity Map Fusion for 3D Model Generation

For reconstructions obtained from classical nadir imagery DSMs or elevation maps are an adequate representation of the scene since limited 3D structure can be extracted as a result of similar viewing directions. However, recently the use of oblique camera systems became more popular. Additional viewing directions at sufficient overlap allow for reconstruction of real 3D geometry as for example façade structure enabling derivation of LOD3 building models, see for example [Verdie et al., 2015]. This also holds for cameras mounted to UAVs possessing oblique viewing directions and, of course, close range imagery. In contrast to 2.5D scenarios data fusion is more advanced because of several reasons. First, geometric properties as ray intersections, image scale and number of observations of the same scene points might largely vary across the scene and the single depth images. This causes varying precisions of points generated for the same surface area. In particular for UAV scenarios additional errors might be introduced due to image blur as a result of limited flight stability. Beside discretization and model errors introduced by stereo matching (pixel locking, foreshortening effects, etc.), additional errors might be present caused by inaccuracies of exterior and interior orientation derived within the bundle block adjustment process. This holds in particular true for UAV imagery, most often captured by consumer grade cameras possessing optics of limited quality and moderate geometric stability. All theses effects lead to inhomogeneous precision of redundant surface observations and an adequate integration strategy has to be provided. The number of error sources is comprehensive and a mathematical model of their collectivity is difficult to obtain. Therefore, inspired by the promising results of algorithms for the generation of elevation data, we propose a median based filter technique for depth map fusion as will be explained in detail in section 6.2.1. Nadir scenarios offer limited variances in image scale and therefore comparable small variations of pixel footprints. With respect to filtering this allows the utilization of a scale-constant data structure, e.g. regular grids. This also implies that neighborhood queries, required for filter operations, are straight-forward to realize. Depth maps derived from close-range and airborne oblique data sets typically possess considerable variances in scale and therefore pixel footprints. To capture variances in surface sampling, multi-scale data structures have to be employed. This can be realized by octree structures as for example utilized in [Kuhn et al., 2014], [Fuhrmann and Goesele, 2011] for which efficient neighborhood queries can be implemented. In this section the algorithms for depth map fusion are explained. First we discuss a novel local approach for the generation of oriented points based on 3D median filtering which are well suited for the derivation of meshes.

6.1 Algorithm Overview

As depicted in figure 6.1 the algorithm can be divided into three main parts: within a first step depth maps are filtered and depth-wise normals are computed. Points of each base image are then streamed into tiles

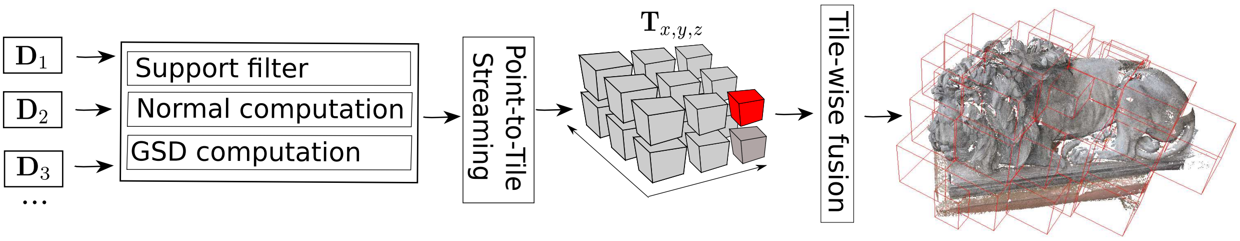


Figure 6.1: Flowchart of the algorithm for the proposed 3D fusion. Entities of each point as normals, pixel footprints and precision values are computed on depth maps. Results are streamed to 3D tiles regularly subdividing object space. Fusion is carried out tile-wise.

subdividing the 3D scene space. Beside the coordinates, normals, image IDs and reliability information are assigned. Then, a median-based filter method is carried out for each of the tiles. Before we discuss single algorithms in sections 6.2 and 6.3 a rough overview is given in the following paragraphs.

Preprocessing of Depth Maps Image-wise filtering of depth observations is based on the reliability which is evaluated by the density of successfully reconstructed depths in their neighborhood (see section 6.2.1). For the subsequent fusion initial normal information of the single points is required. Instead of normal computation in 3D space we prefer to compute normal information directly in the depth images and store this information along the coordinates, see sections 6.2.2. Computation of normals in image space offers a number of advantages. First, neighborhood queries are fast and easy to implement. Second, the set of neighbors to be incorporated for 3D normal computation is challenging for scale variant data sets. This holds particularly true if data points are noisy as for points generated by multi-view approaches. Our algorithm for normal computation is based on a restricted quadtree as discussed in section 5.4.1. Thereby a geometry-adaptive triangulation is derived from a depth map, from which then normals can be extracted. Furthermore, in order to support adaptive sampling, the depth-wise pixel footprints are computed.

Subdivision in Tiles Due to the considerable sizes of airborne data sets scalability has to be ensured. As for the generation of elevation maps the input of the algorithm is defined by 3D cubes which regularly subdivide object space. Each of these 3D tiles contain all points generated by the multi-view matching approach located in the respective spatial extend, therefore points in the single tiles are results of different depth maps across the camera network. The reason for tile-wise processing is twofold. First and as mentioned before, this guarantees scalability to large datasets. Second, this enables convenient high-level parallelization of processing single data chunks.

Median-Based Fusion The actual fusion is then performed on each of the single tiles. The general idea of the approach is to median filter points along the surface normal. Thereby the initial surface is given by the set of points $\mathcal{P}$ across all depth maps possessing the locally smallest pixel footprints. This not only ensures adaption to the finest surface sampling, but also, since precision is correlated to the GSD, favors most precise points to be part of the initial set of surface points. Eventually, as described in section 6.3, the set of $\mathcal{P}$ is filtered iteratively along the point normals.

6.2 Preprocessing of Depth Maps

In this section we discuss the algorithms for computation of point normals and the pixel-wise reliability measure carried out prior to the actual fusion. All algorithms operate on the depth images derived by the multi-view correspondence linking approach discussed in section 4.

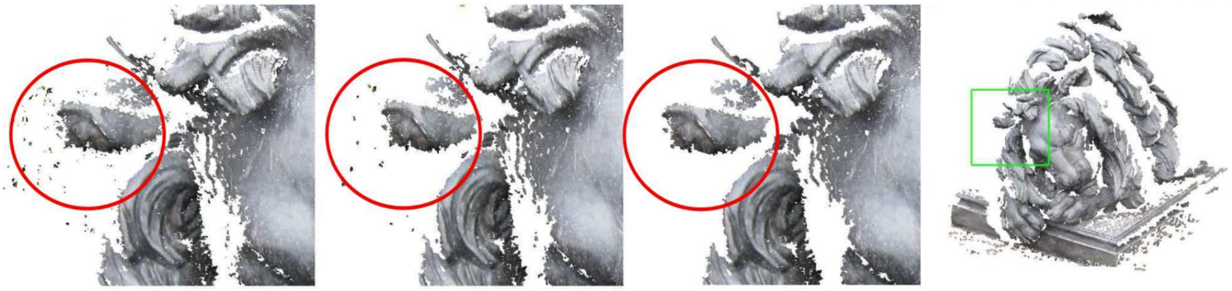


Figure 6.2: Point clouds corresponding to the filtered depth maps for different thresholds t_s. Figures in the upper row are detailed views of the area depicted by the green rectangles. Left: $t_s = 0$; Middle:$t_s = 21$; Right:$t_s = 41$.

6.2.1 Outlier Filtering Based on Support Images

Although multi-view approaches may result in rather clean point clouds, not all erroneous observations can be detected based on the evaluation of geometric and/or radiometric consistency across multiple views. One major problem are regions of object borders in combination with low-textured background (see the red circle in figure 6.2). In these parts pixel consistency measures are not distinctive and depth estimation is mainly powered by regularization/smoothness constraints. As a result edges are over-matched what cannot be reliably detected by consistency constraints across the images. These erroneous depths are typically represented by small-sized pixel patches. Hence, the presented filter approach is based on the reconstruction density in the local neighborhood of single depth observations. This measure will be referred to as support in the following text. Inspired by the path-wise accumulation process of SGM, for each pixel $\mathbf{x}(x, y)$ the support is computed by the evaluation of consistency with its proximate pixels along 16 image paths

$$S(\mathbf{x}(x,y)) = \sum_{i=1}^{16} s_i(\mathbf{x}(x,y)) \tag{6.1}$$

The pixel- wise support values are computed by evaluation of subsequent pixels $\mathbf{x}_n, \mathbf{x}_{n+1}$ along each path i according to

$$s_i(\mathbf{x}) = \sum_{n=0}^{m} T(\mathbf{x}_n, \mathbf{x}_{n+1}) \tag{6.2}$$

thereby n denotes the offset with respect to $\mathbf{x}$ along the path direction and m marks the position of the closest invalid pixel along this path direction. $T(\times)$ is an operator evaluating to 0 or 1 according to

$$T(\mathbf{x}_n, \mathbf{x}_{n+1}) = \begin{cases} 1, & \text{if } d(\mathbf{x}_n), d(\mathbf{x}_{n+1}) \textit{ valid} \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

If the final support $S(\mathbf{x}(x, y))$ is below a certain value t_s its corresponding depth will not be considered for further processing. Figure 6.2 displays results for different thresholds t_s. The utilized dataset consists of 57 consumer-grade images depicting a lion statue publicly available at [Jancosek and Pajdla,]. The red circles mark areas of erroneous depths due to overmatching. For a threshold of $t_s = 21$ we found overall satisfying point filtering within our tests. We used this threshold for all of our experiments.

6.2.2 Normal Computation using Restricted Quadtrees

Depth-wise normals are computed by triangulation of depth maps using a RQT approach. Based on the faces normals can then be extracted. Similar to the triangulation of elevation data in this work we implemented a top-down RQT construction strategy for the triangulation of depth maps. Thereby initially a depth map is represented by a single node. Nodes (depths) are added recursively if error criteria are not fulfilled, i.e. the node to be added represents actual geometry. Recursively, in a top-down manner, all nodes are checked for error criteria based on their two dependent nodes and activated if these are fulfilled. If a node is evaluated to be a part of the triangulation then its dependent nodes and their dependencies have to be activated too. From the set of finally activated nodes a matching triangulation can be extracted.

The error criterion describes the error which is introduced by not adding a vertex to the triangulation, thus controlling the degree of simplification. We base our dynamic error criterion on a depth observation-wise precision approximation $\epsilon(\mathbf{x})$. To this point $\epsilon(\mathbf{x})$ is based on the pixel footprint $f(\mathbf{x})$, approximating the precision of depth observations by their sampling rate on the surface. Let n be a node with the depth $d(x, y)$ and its dependent nodes n_1, n_2 with $d_1(x, y)$ and $d_2(x, y)$ respectively (figure 5.11). Each depth point is assumed to be measured with a uncertainty of $\epsilon(\mathbf{x}) = f(\mathbf{x}) * t_q$ with the constant threshold t_q. The node n is inserted to the triangulation if it is not located in the local noise band which is defined by $d_1 \pm \epsilon_1$ and $d_2 \pm \epsilon_2$. Similar to the mesh extraction of elevation data, a depth value is considered to contribute to the geometry if equation 5.3 is fulfilled.

Once all depth observations have been checked for the error criterion the RQT construction is completed and a triangulation can be derived. For each vertex $\mathbf{v}(\mathbf{x})$ which is element of the triangulation a normal $\mathbf{n}(\mathbf{x})$ is computed. It is determined by the average of all normals of faces possessing the respective vertex as a corner. For depth observations which were neglected during RQT construction, in other words vertices $\mathbf{v}(\mathbf{x})$ which are not element of the triangulation, the normal is interpolated using the normals of the face covering the coordinates $\mathbf{x}$. Eventually, in order to reduce input/output loads the normals are encoded as 16 bit integers following [Meyer et al., 2010] as [Cigolle et al., 2014] found this method giving good results in terms of accuracy and speed.

6.2.3 Computation of local GSDs

The footprint of a pixel on the object is an important information not only within the RQT-based normal computation but also within the fusion process. First, it is used to control the local sampling density during fusion. Second, since representing image scale, the pixel footprint can be used to draw conclusions concerning the relative precision between redundant observations. Given the depth d and the focal length c_{pix} in pixel units the pixel footprint $f(\mathbf{x})$ is easily calculated as

$$f(x, y) = \frac{d}{\sqrt{c_{pix}^2 + x^2 + y^2}}. \tag{6.4}$$

In practice, instead of storing the GSD along point coordinates, the image ID for each point is stored and pixel footprints are computed when loading points in the fusion step. The parameter d can be derived by computation of the distance points and the respective camera center.

6.3 Median-Based Filtering

After streaming point coordinates along their attributes to their respective tiles, eventually the median-based fusion is carried out. First, we explain how we extract an initial point set $\mathcal{P}$ representing the point set offering the smallest sampling rates on the surface. Therefore, we use a multi-scale octree structure into which all points of a single tile are inserted. Point coordinates and the point-wise pixel footprint define the octree cell in which an observation is located in. Points located on the lowest level (leaf nodes) of the octree then indicate surface samples offering densest sampling and in general best precisions. Next, the set $\mathcal{P}$ is median filtered along the point-wise normals in an iterative manner.

Linear Octrees

Similar to the restricted quadtrees discussed in section 5.4.1 octrees are data structures separating space by regular boxes. In contrast to 2D quadtees they are separating 3D space by cubes. Each of the cubes contains

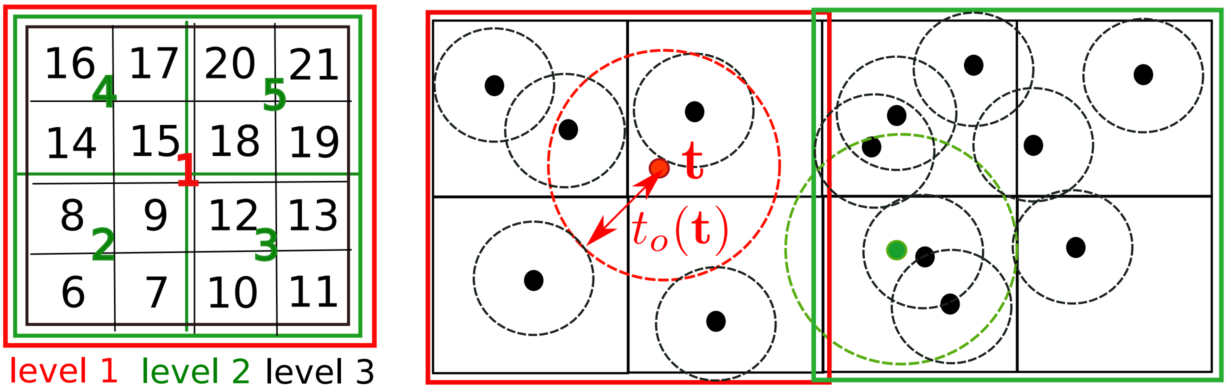


Figure 6.3: Left: Definition of box indexes for a quadtree. Right: Visualization for the criterion assigning points to the octree. Circles mark the point-wise GSD, points represent the coordinates. Points are assigned to the box they are located in and the side lengths of the box are comparable to the GSD.

8 daughter cubes regularly subdividing its mother cube. Points to be inserted are assigned to cubes (nodes) if certain criteria are fulfilled. For example a point is assigned to the smallest box fully containing a sphere around the respective point coordinates. Hence, the octree structure implements an ordering scheme of points providing queries for point location as well as neighborhood queries. Our implementation follows the algorithms proposed by [Gargantini, 1982],[Press et al., 2007]. For tree traversal the link between mother nodes and daughter nodes have to be provided. Instead of a more conventional implementation where links between mother nodes and daughter nodes are realized by doubly linked pointers we prefer an implementation based on hash maps, also called Linear Octrees. The main advantage is that the usage of pointers (each 64bit on common hardware) storing mother-daughter and daughter-mother relationships is avoided. This reduces memory requirements, which supports the processing of single tiles in parallel and therefore speeds up the integration process. Since regularly dividing space, the single boxes in a octree can be numbered by a unique scheme. Figure 6.3 (left) gives an example for box indexing in two dimensions, e.g a quadtree structure. This indexing enables traversal of the tree: given a specific box index i the indexes m to a mother box and d_l to the leftmost daughter bos can be conveniently computed following

$$m = \frac{i+2}{4} \tag{6.5}$$

$$d_l = 4(i-1)+2. \tag{6.6}$$

This indexing scheme can be easily adapted for 3D cubes. The formulas of m and d_l are then given by

$$m = \frac{i+6}{8} \tag{6.7}$$

$$d_l = 8(i-1)+2. \tag{6.8}$$

This allows for navigation in the octree. For typical data sets derived from image matching not each of the boxes is polluted but only boxes close to the actual object surface. To speed up traversal the information whether a box is polluted and pollution of its daughters is stored by a hashmap. The key of the hash map is the box index and its value is an integer holding the information about the pollution status. This integer is designed on binary level, where the first bit decodes if the box is polluted and bits 1 to 8 decode the pollution of daughter boxes or if it is a mother to any polluted box. This allows for quick tree traversal since boxes which do not contain points themselves nor does any daughter can be neglected. Note that equation 6.7 and the hashmap fully substitute the double mother-daughter and daughter-mother links. Secondly a data structure holding the information of the actual geometric entities contained by the single boxes has to be provided. This is realized by a multi hashmap with the key being the box index and its list elements being the contained geometric entities.

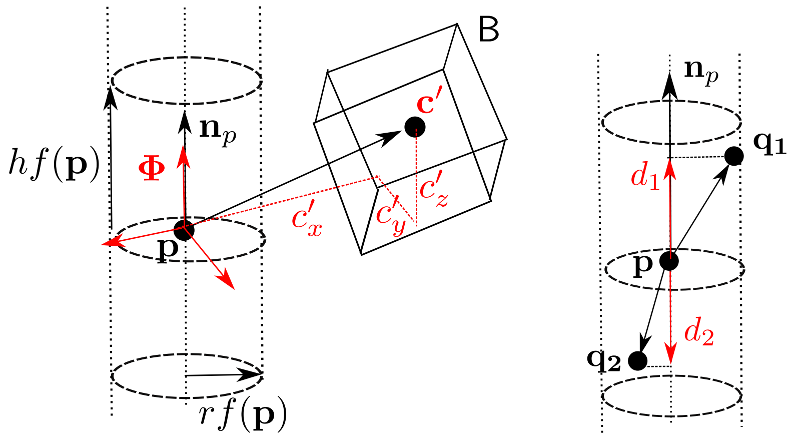


Figure 6.4: Left: Evaluation of the criterion describing if an octree box contains any points inside a specified cylinder $\mathbf{p}$, $\mathbf{n}_p$, $hf(\mathbf{p})$, $rf(\mathbf{p})$. Conditions are based on the coordinates c'_x, c'_y, c'_z of box center $\mathbf{c}'$ defined w.r.t. the coordinate system Φ. Right: Median filtering along the point normal $\mathbf{n}_p$. Translations induced by candidates $\mathbf{q} \in \mathcal{Q}$ are given by their projection on the cylinder axis $\mathbf{n}_p$.

Derivation of the Initial Point Set $\mathcal{P}$

To identify the point set $\mathcal{P}$ we sort all points $\mathbf{t} \in \mathcal{T}$ contained in a single tile into the octree. Let B be an octree box with the side length s. As visualized in figure 6.3 (right), a point $\mathbf{t} = [t_x, t_y, t_z]$ with the footprint $f(\mathbf{t})$ is located in B if the point is inside the cell and B is the smallest box satisfying

$$s > t_o(\mathbf{t}) \tag{6.9}$$

with

$$t_o(\mathbf{t}) = \alpha \frac{f(\mathbf{t}) + \beta \frac{1}{n} \sum_{\mathbf{t} \in \mathcal{T}} f(\mathbf{t})}{1 + \beta}. \tag{6.10}$$

To be able to extenuate the influence of single footprints $\mathbf{t}_o(\mathbf{t})$ is composed of the local footprint and the average pixel footprints in the tile or the whole data set (n is the number of considered samples). For large β a uniform sampling can be derived neglecting scale variances across single observations completely. The parameter α controls the sampling density of the surface, therefore using a large valued α the surface is undersampled and for a small valued α oversampling is enforced which might be desirable for high redundant datasets. After sorting all points of a tile to the octree the initial point set $\mathcal{P}$ is derived by identifying all leaf nodes. Per leaf node one point is generated by averaging coordinates and normals of all contained points.

Median filtering - First Iteration

The point set $\mathcal{P}$ comprises points which are reconstructed possessing the smallest pixel footprint within a local neighborhood. However, errors resultant from registration and propagated from dense stereo, as well as properties of ray intersection angles are not modeled by the pixel-wise footprints. Generally this causes the extracted points $\mathcal{P}$ being noisy, hence we median filter the points set $\mathcal{P}$ along the surface normal. Thereby candidates incorporated into the filtering are all points stored in the leaf nodes of the octree. Note that this set of leaf node points in general is much larger than $\mathcal{P}$. Within a first step for each $\mathbf{p} \in \mathcal{P}$ a set of neighboring points is derived from the octree. Paying respect to the lager uncertainty in direction of the surface normal as well as to outliers, the neighbors are defined by the set of points $\mathcal{Q}$ located in a cylinder with its central axis given by $\mathbf{p}$ and its normal $\mathbf{n}_p$. The cylinder radius and height are dependent of the footprint and specified by $rf(\mathbf{p})$ and $hf(\mathbf{p})$ receptively (see figure 6.4 (left)). To limit smoothing and artifacts at tile borders we choose a rather small radius $r = 1.2$. The tube height in our experiments is set to $h = 10$.

Identification of the point set $\mathcal{Q}$ involves nearest neighbor queries on the octree structure. Starting at the mother node each octree cube is checked if itself or any daughters might contain leaf node points located in the cylinder of question. This query is checked frequently thus has to be designed carefully. Let B be a candidate octree box with the center $\mathbf{c}$ and the side length s. Moreover, let $\mathbf{p}$, $\mathbf{n}_p$, r, h define the cylinder (see figure6.4 (left)). We construct a Cartesian coordinate system Φ with the origin in $\mathbf{p}$ and the z-axis pointing in direction of $\mathbf{n}_p = [n_x, n_y, n_z]$. The axes of the coordinate system are defined by the columns of the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix} = \begin{bmatrix} 1 & 1 & -\frac{n_x+n_y}{nz} \\ & (\mathbf{r}_1 \times \mathbf{r}_3)^\top & \\ n_x & n_y & n_z \end{bmatrix} \tag{6.11}$$

The box center can then be transferred into the coordinates system Φ by

$$\mathbf{c}' = \mathbf{R}(\mathbf{c} - \mathbf{p}). \tag{6.12}$$

The octree box B may contain points located in the cylinder if

$$\sqrt{(c'_x)^2 + (c'_y)^2} < r + \sqrt{3}s \tag{6.13}$$

and

$$c'_z < h. \tag{6.14}$$

The term $\sqrt{3}s$ in equation (6.13) represents the radius of a sphere enclosing B. If conditions (6.13) and (6.14) are not fulfilled the traversal of daughter nodes is terminated. If the conditions are fulfilled and additionally the examined box B is a leaf node all points contained by B are a subset of $\mathcal{Q}$.

Once the set of neighbors $\mathcal{Q}$ is identified all $\mathbf{q} \in \mathcal{Q}$ are checked to be located the specified cylinder. This is done following equations (6.12)-(6.14) by exchanging roles of box centers and points. If not located in the tube the sample is removed from $\mathcal{Q}$. Additionally for each $\mathbf{q} \in \mathcal{Q}$ the angle between its normal $\mathbf{n}_q$ and $\mathbf{n}_p$ is computed. If this angle is larger than 60° the sample is removed from $\mathcal{Q}$ and discarded for further filtering. This way incorporation of points not representing the same surface is avoided. After removal of suspicious samples the actual filtering is performed. The basic idea of the implemented median filtering is to translate the coordinates of $\mathbf{p}$ along its normal $\mathbf{n}_p$. The translation is given by the median of translations induced by $\mathbf{q}_i$. More precisely, a translation of a sample $\mathbf{q}_i$ with respect to Φ is given by its projection onto $\mathbf{n}_p$

$$d_i(\mathbf{q}_i) = [\mathbf{q}_i - \mathbf{p}]^\top \mathbf{n}_p, \tag{6.15}$$

see figure 6.4 (right). Then the updated coordinates $\mathbf{p}'$ are computed as

$$\mathbf{p}' = \mathbf{p} + \mathbf{n}_p \; median[d_i(\mathbf{q}_i)]. \tag{6.16}$$

Median Filtering - Additional Iterations

The median-based integration described before enforces the set $\mathcal{P}$ to converge to the median surface. However, for noisy data sets multiple iterations might further improve the final surface. Recall that within the first iteration all points contained in the all leaf nodes of the octree were considered for integration. To speed up further iterations we restrict filtering on points $\mathbf{p} \in \mathcal{P}$ solely. As before, $\mathcal{P}$ has to be sorted to a 3D data structure enabling cylinder-based neighborhood queries. An octree as presented in the last section would be suitable for this task. However, we found that for our data sets these queries can be processed faster using KD-trees.

KD-trees are structures partitioning k-dimensional points based on half spaces. Since in this work we are interested in 3D entities we restrict the following discussion to 3D space. Initially the first two points to be inserted are divided by a plane perpendicular to the axis of the first dimension (x-axis). These spaces define the initial nodes of the tree. For a new point $\mathbf{p}_{new}$ to be inserted the leaf node in which $\mathbf{p}_{new}$ is located in along with the point $\mathbf{p}_{old}$ already contained by the node are identified. The node then is further split

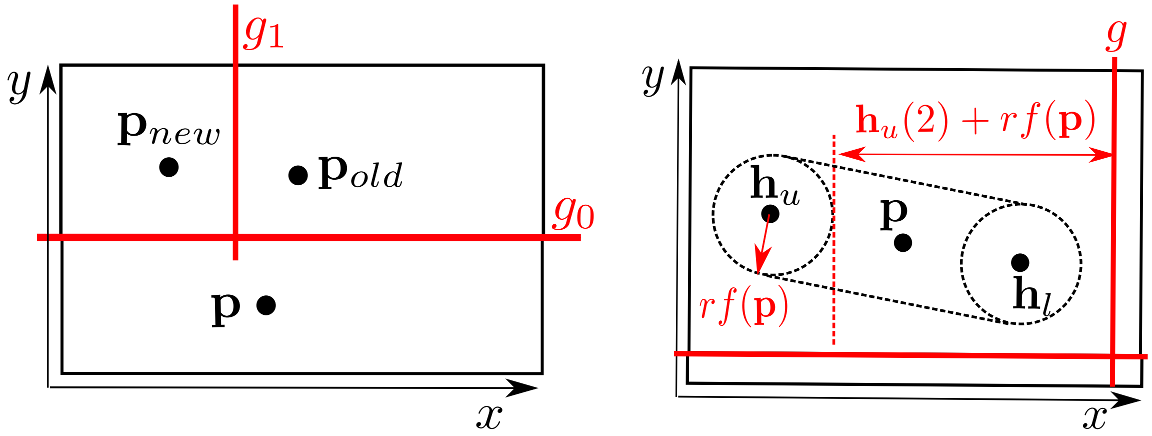


Figure 6.5: Left: Partitioning scheme of a KD-tree. Each node contains one point, nodes are divided by hyperplanes g_i. Direction of planes are incremented over the dimension D of space $\mathbb{R}^n$ (in this example $\mathbb{R}^2$). Right: Evaluation of the criterion describing if a KD-tree node fully contains a given cylinder. Therefore, the D-th component of the coordinates $\mathbf{h}_u(D) + rf(\mathbf{p})$, $\mathbf{h}_l(D) + rf(\mathbf{p})$ have to be completely located on one side of the plane g.

by a plane separating $\mathbf{p}_{new}$ and $\mathbf{p}_{old}$ leading to two new leaf nodes, see figure 6.5 (left). The orientation of the plane is defined orthogonal to the second dimension. Subsequent insertions are performed in a similar way: first the leaf node, the contained point and the dimension d defining the direction of its last separation are identified. Then, the node is divided by a plane possessing a normal in direction of the incremented dimension $((d+1) \mod 3)$, resulting in two new leaf nodes. For implementation details the reader is referred to the algorithm proposed in [Press et al., 2007].

In order to extract the point set $\mathcal{Q}$ containing the points located in a cylinder defined by the point $\mathbf{p}$, its axis $\mathbf{n}_p$, its radius $rf(\mathbf{p})$ and height $hf(\mathbf{p})$ efficient neighborhood queries have to be provided. To identify the node whose daughters hold all points in the tube we identify the node fully containing the tube. Therefore, we construct two points $\mathbf{h}_l = \mathbf{p} - \frac{h}{2}\mathbf{n}_p$ and $\mathbf{h}_u = \mathbf{p}_l + \frac{h}{2}\mathbf{n}_p$, see figure 6.5 (right). Starting at the root of the tree for each node it is evaluated if the two spheres defined by $\mathbf{h}_l$ and $\mathbf{h}_l$ and the radius $rf(\mathbf{p})$ are fully contained by one of the daughter nodes. Let g be the position of the plane separating the daughter nodes in dimension D. Then the conditions are given by

$$\mathbf{h}_{l,u}(D) + rf(\mathbf{p}) < g \tag{6.17}$$

and

$$\mathbf{h}_{l,u}(D) - rf(\mathbf{p}) > g \tag{6.18}$$

If a node fulfills both conditions (6.17),(6.18) and its daughter nodes do not, the node is guaranteed to be the smallest node holding all points located in the cylinder. Once this node is identified, all points contained by the daughters are skimmed and those located in the cylinder define the set $\mathcal{Q}$. Analogously to filtering in the first iteration, all inconsistent samples $\mathbf{q}_i$ possessing normals largely differing from $\mathbf{n}_p$ are identified and removed from $\mathcal{Q}$. The remainder is median filtered along the point normal $\mathbf{n}_p$ as discribed in the previous section.

6.4 Visibility Check

In order to filter further suspicious points we eventually perform a visibility check. For each tile $\mathbf{T}$ the points $\mathbf{p}_i$ produced from view V_i are projected to a virtual image plane defined by its exterior and interior orientation. Thereby two virtual images are generated: a virtual depth image $\mathbf{D}_i$ and a virtual image $\mathbf{S}_i$ recording the support of the points. Once, these virtual images are derived, all $\mathbf{q} \in \mathbf{T}$ are checked if they occlude the surface represented in $\mathbf{D}_i$. Since erroneous points $\mathbf{p}_i$ behind the actual surface would lead to erroneous invalidations we include the point-wise support within our consistency check. Let $\mathbf{D}_i(x, y) = d_i$ be a depth in the virtual depth image and $\mathbf{S}(x, y) = s_i$ the support respectively. Furthermore, let $d_\mathbf{q}$ be the depth induced by projection of the point $\mathbf{q}$ onto the pixel (x, y) of view i with the support $s_\mathbf{q}$. Then, the point $\mathbf{p}_i$ is invalid if

$$d_i + hf(\mathbf{p}) > d_\mathbf{q} \quad \text{and} \quad s_{qmat} > 2s_i. \tag{6.19}$$

Else the point $\mathbf{q}$ is considered valid.

6.5 Results and Discussion

In this section the proposed algorithm for fusion is evaluated. For accuracy and completeness evaluations we use well established multi-view benchmarks for close range reconstruction. As for the evaluation of algorithms for depth map generation we first utilize Strecha's Fountain dataset and additionally the Middlebury MVS evaluation. In contrast for aerial oblique datasets, meaningful ground truth is difficult to obtain. However, we show the capability of processing city-scale projects by a more quantitative evaluation of results using an airborne oblique image block.

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,3}(P)$	-0.92	1.69	1.92	70.52	-0.85	1.40	1.64	68.68
$\mathbf{I}_{b,4}(F)$	-0.95	1.61	1.87	70.59	-0.88	1.31	1.57	68.79
$\mathbf{I}_{b,5}(F)$	-0.94	1.59	1.85	70.39	-0.86	1.25	1.51	68.43
$\mathbf{I}_{b,6}(F)$	-1.00	1.63	1.92	70.28	-0.90	1.25	1.54	68.15
$\mathbf{I}_{b,7}(P)$	-1.08	1.73	2.04	68.21	-0.97	1.31	1.63	66.00
$\mathbf{I}_{b,8}(P)$	-1.27	1.85	2.25	60.62	-1.14	1.44	1.83	58.42

Table 6.1: Statistics for differences of ground truth depth maps and re-projected point clouds obtained by the proposed fusion approach. All values are expressed in [pix] except of $\%_{val}$ and $\%_{val,3}$ [%].

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,3}(P)$	-1.08	1.68	2.00	68.63	-0.98	1.42	1.72	66.67
$\mathbf{I}_{b,4}(F)$	-1.29	1.63	2.08	69.15	-1.18	1.35	1.80	67.17
$\mathbf{I}_{b,5}(F)$	-1.05	1.68	1.98	69.54	-0.95	1.37	1.66	67.67
$\mathbf{I}_{b,6}(F)$	-1.02	1.63	1.92	66.40	-0.91	1.27	1.56	64.50
$\mathbf{I}_{b,7}(P)$	-0.68	1.69	1.83	60.77	-0.58	1.37	1.49	59.34
$\mathbf{I}_{b,8}(P)$	-1.05	1.85	2.13	47.73	-0.94	1.48	1.75	46.26

Table 6.2: Statistics for differences of ground truth depth maps and depth maps obtained by the multi-view stereo. All values are expressed in [pix] except of $\%_{val}$ and $\%_{val,3}$ [%].

6.5.1 Fountain

To evaluate the accuracy and show that precision of 3D point clouds can be improved using the proposed depth map fusion we utilize Strecha's Fountain data set. Therefore, we matched each of the i base images I_i against the 4 closest match images and produced point tiles by the multi-view correspondence linking approach explained in section 4.3. Resulting points then were fused. In order to compare the results to the ground truth depth maps all triangulated points were re-projected to generate virtual depth images V_i. Since the fused cloud represents 3D geometry, points corresponding to occluded surfaces have to be identified and removed. Therefore, if multiple point map to the same pixel we only consider the closest one. Moreover, if differences of reconstructed and ground truth depths differ by more than 5cm (> 10 times the average pixel footprint) the point is not incorporated into the evaluation. Note that still some depths corresponding to occluded surface patches may be incorporated and evaluated precision of fused clouds may be better than stated within this test. Table 6.5 depicts the obtained statistical values for deviations of virtual depth maps and ground truth depth maps expressed in pixel units (except of completeness measures) following the same definitions as in section 4.4.2. Corresponding metrical values can be found in the appendix 8.4. As before a systematic shift, most probably due to inaccurate image orientation, can be observed. Small deviations between σ and σ_3 indicate that a large portion of gross blunders were removed in the 5cm filter step. σ_3 standard deviations of all virtual depth maps are below 1.5 pixels. To manifest improvements of the the proposed fusion, the same characteristics of multi-view depth maps D_i, filtered using the same $5cm$ criterion (see table 6.5) were computed. The chart 6.7 depicts the differences of σ_3 in pixels and clarifies that improvements up to 0.13 pixels can be obtained. However, these results may be biased by occluded patches as mentioned before. As expected the percentage of reconstructed pixels is higher for the fused clouds, since data holes are compensated by matches from neighboring views (see chart 6.6). Within the fusion process the number of points was reduced by 84.24%. Time consumption amounted 2.16 minutes on a six core Intel I7 clocked at 3.3GHz.

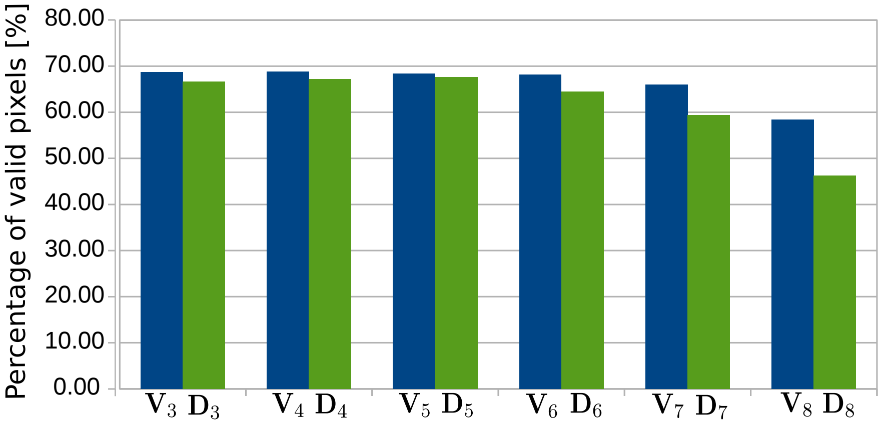


Figure 6.6: Percentage of matched pixels w.r.t all image pixels. Blue: virtual depth maps derived from fused points. Green: original depth maps without fusion.

6.5.2 Middlebury Benchmark

The second evaluation is carried out on the Middlebury MVS benchmark data set [Seitz et al., 2006]. Therefore, we generated depth maps for the temple and dino datasets in 'full' image configurations. Within depth map generation each image is treated as reference image which is stereo matched against the 8 closest neighboring views. By multi-view forward intersection and constraints on geometric consistency redundant observations across the set of disparity maps are refined resulting in one depth map per view. Since the evaluation is performed on triangle meshes, we extract oriented points using the presented approach and subsequently apply a Poisson reconstruction [Kazhdan et al., 2006]. In order to extract a surface close to the oriented points generated by our approach an octree depth similar to our octree is specified and no constraints regarding the minimal number of points per octree cell is applied. Nevertheless, the Poisson reconstruction imposes additional smoothing. Figure 6.8 depicts the results for the two data sets. Median filtering in the kd-tree stage was carried out as described in section 6.3 using 2 iterations. Since depth observations of the dino dataset turned out to be more noisy, the parameters for the tube radius was set to $r = 4$ instead of $r = 1.2$ for the temple dataset. The dynamic sampling width for both data sets were set to $\alpha = 2$ and $\beta = 0$ (see equation (6.10)). The first column in figure 6.8 shows the points contained by all refined depth maps. The second column depicts the oriented point set derived by the proposed algorithm. As can be seen the gross of outliers are removed and small details are preserved: for the temple data set 99.5% of observations possess an average deviation of 0.55mm to the ground truth and 0.47mm for the dino

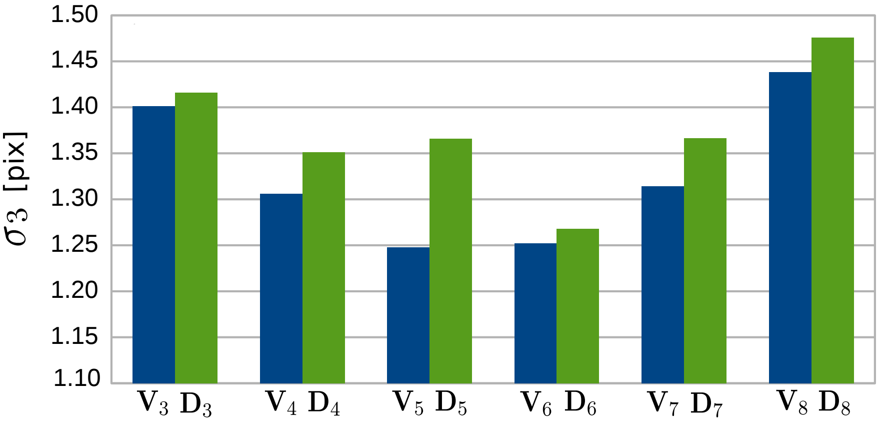


Figure 6.7: σ_3 based on differences between depth maps and ground truth depth maps. Blue: Virtual depth maps derived from fused points. Green: original depth maps without fusion.

data set respectively. The actual pixel footprints are in the range of 0.1-0.4mm. Dependent on accuracy and completeness levels given on the Middlebury evaluation page, the algorithm ranks in the range of 13 to 1 (with several others), therefore can be considered state of the art. The run times for the fusion process on a single core clocked at 3.3 GHz amounted 171 seconds for the dino and 78 seconds for the temple data set.

6.5.3 Airborne Oblique Dataset

As discussed in the last section the proposed algorithm delivers state-of-the-art results regarding completness and precision. In this section we additionally evaluate the capability for processing large scale data sets, in particular airborne oblique imagery. Furthermore, a more detailed analysis of the single filtering steps is given. Therefore, imagery captured by an IGI Penta Cam is utilized. This system is composed of a nadir camera and four camera heads possessing inclined viewing directions (35°, forward, backward, right, left). Imagery from the nadir flight provide overlaps of 80% in flight direction and 60% side overlap. For the obliques this results in overlaps of approximately 65% to 80%, depending on the image region. Due to the slanted viewing directions GSD are inhomogeneous, the average pixel footprint amounts 0.14 meters. Oriented points were generated by matching each image against its closest neighboring images. An image is treated as an slave image if its overlap with the master image amounts at least 20%. After pairwise matching, points are triangulated, normals are computed and streamed to tiles with side lengths of 30 meters. The fusion was parametrized using the standard tube radius $r = 2$ and tube height $h = 10$. The sampling parameters were set to $\alpha = 1$ and $\beta = 0$, the number of iterations kd-tree based filtering was specified as 2. Figure 6.9 and 6.10 depict the original and fused points after octree and kd-tree stages. It becomes clear that in the octree stage normals are rubustified and the number of points is greatly reduced (in this example by 84%). However, scene coverage remains close to constant, meaning only limited data holes are introduced. Moreover, small details as poles and thin structures are preserved. In figure 6.10 (e),(f) the benefits of the additional kd-filter steps are visualized. Additional blunders which were not filtered in the octree stage are removed because of improved normal information and normals are further robustified. Although for the present dataset these effects are limited, iterative kd-filter steps become more important if the quality of image orientation is limited. The processing time for 40 cubes possessing side lengths of 30 meters amounted 8.51 minutes utilizing an Intel I7 clocked at 3.3 GHz.

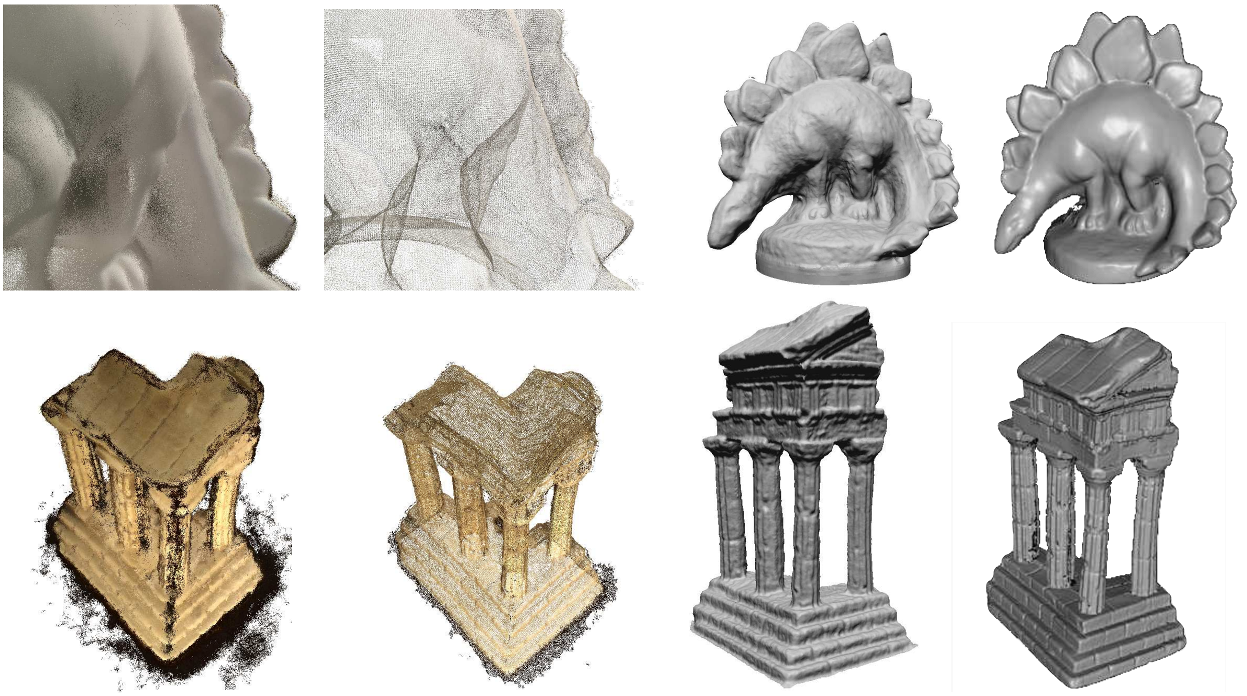
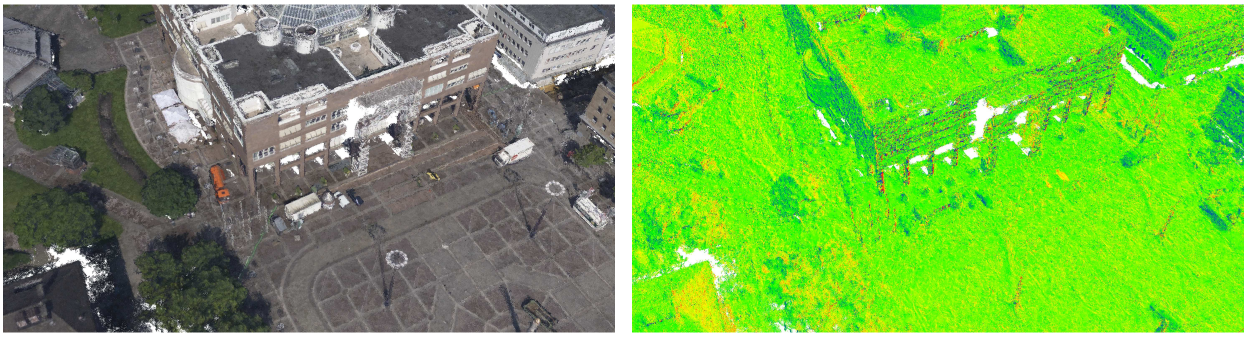


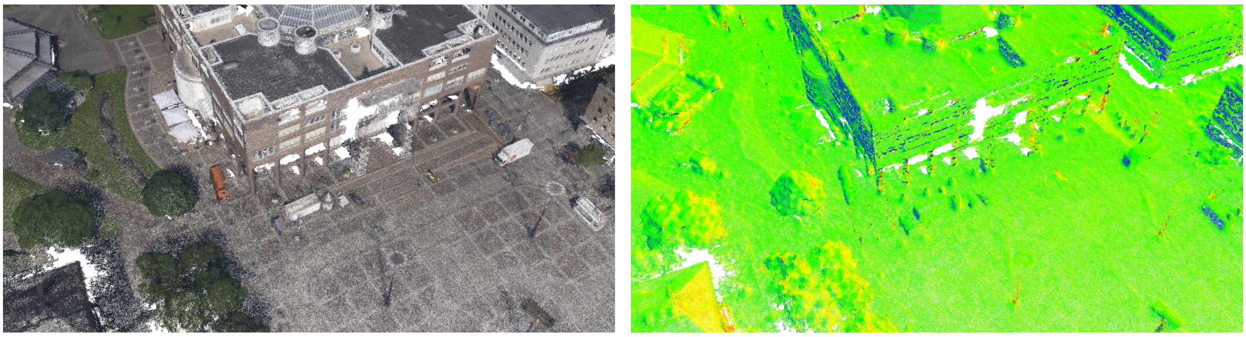
Figure 6.8: Results for the temple and dino data sets. From Left to right: raw point cloud; oriented points resulting from the our method; mesh generated by Poisson reconstruction using our oriented points; ground-truth mesh.

6.6 Mesh Extraction

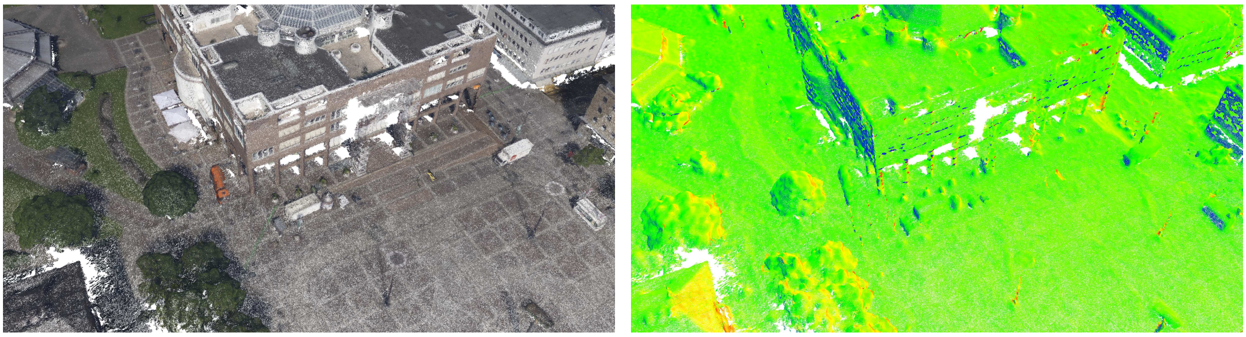
With the availability of data representing real 3D structure, also meshes as representation for surfaces gain more popularity for airborne applications. Beside commonly used for visualization purposes meshes offer two main advantages compared to point clouds. First, neighborhood information is stored along the point data supporting classification, analytics, and editing tasks. Second, data can be compressed to a high degree by representing areas with limited geometric variances by single triangles. In this section we show that oriented points produced either by the implemented fusion method or raw point clouds along the extracted normals can be utilized to derive meshes of impressive quality. Based on the generated oriented points, we apply the Poisson reconstruction as proposed in [Kazhdan et al., 2006] and [Kazhdan and Hoppe, 2013]. Whilst uncritical for most moderate sized close range data sets, the algorithm offers only limited scalability which becomes prohibitive for city scale airborne data sets. To overcome this restriction we use a tile-based scaling scheme merging neighboring point cubes produced by the fusion approach, in the following referred to blocks. To guarantee matching geometry between neighboring blocks some overlap o is specified (in the present example 30 meters). Figure 6.11 depicts a mesh extracted from the oblique image data set described in section 6.5.3. The extend of the reconstruction amounts 800 by 1200 meters at an average GSD of 14cm which clarifies applicability for processing large scale scenes. Figure 6.12 shows an area at block borders and clarifies that tiling effects are not present and geometry in transition areas are close to identical. To demonstrate the achievable precision, 3D mesh results are visually compared to results derived by the reconstruction algorithm proposed for 2.5D surfaces. The first row in figure 6.13 depicts a central area of the block reconstructed by raw non-filtered points obtained by stereo matching and subsequent multi-baseline forward intersection. As can be observed the meshes extracted by the Poisson reconstruction (a) yield smoother surfaces preserving marginally less amount of detail. This indicates that the proposed algorithms do not fully exploit the given redundancy and precision of meshes could be further improved. However, in comparison to the 2.5D algorithm extracted meshes are topologically correct (also in presence of data holes)



(a) RGB - all points *(b)* Normals - all points

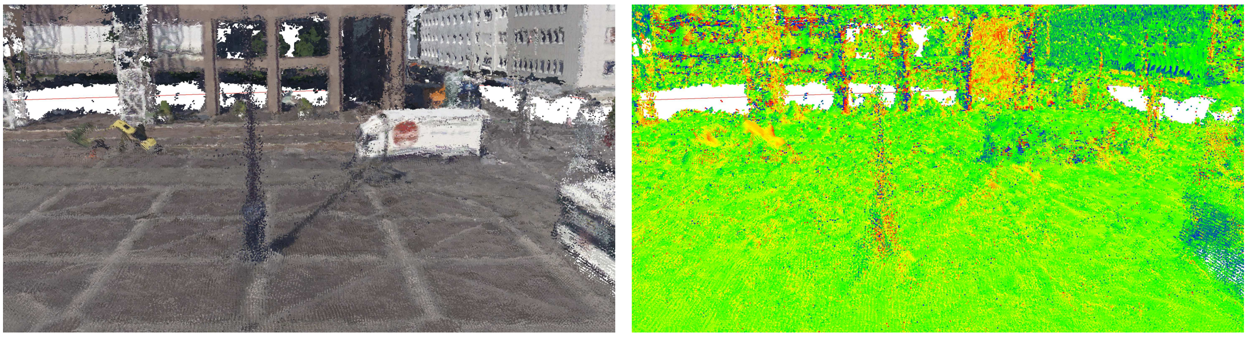


(c) RGB - points after octree stage *(d)* Normals - points after octree stage

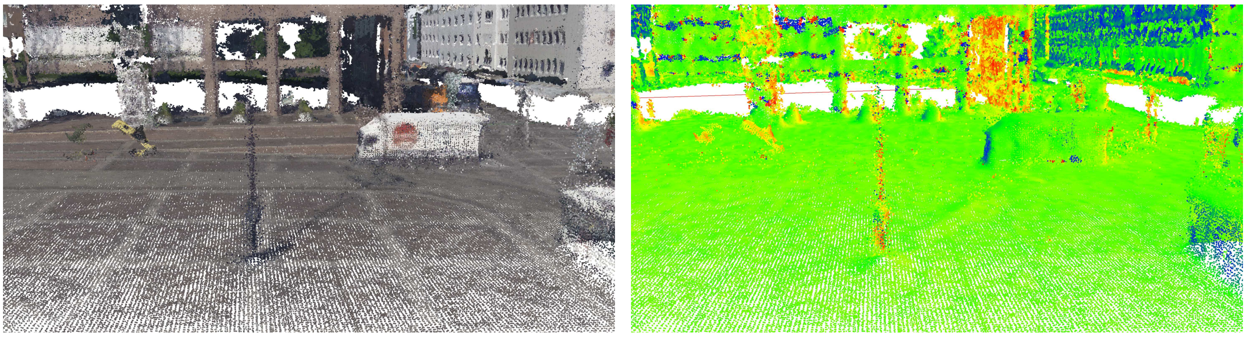


(e) RGB - points after kd-tree stage *(f)* Normals - points after kd-tree stage

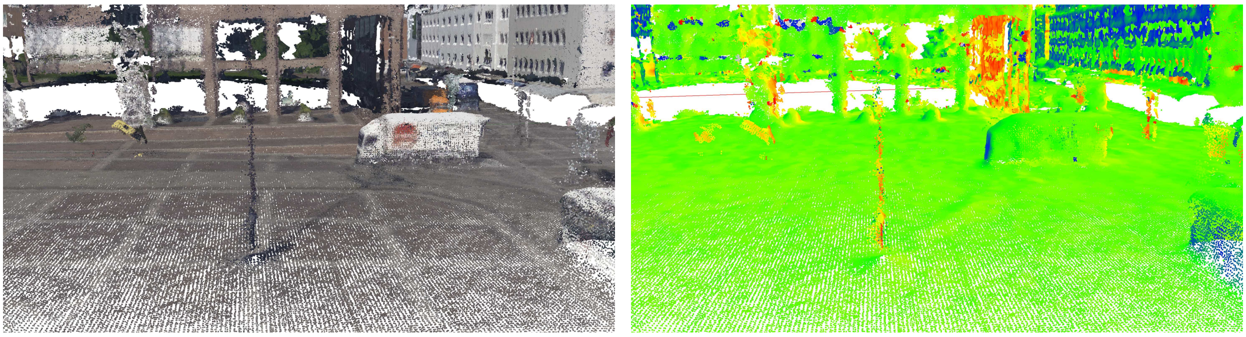
Figure 6.9: Visualization of raw point clouds and point clouds after octree and kd-tree stages.



(a) RGB - all points *(b)* Normals - all points



(c) RGB - points after octree stage *(d)* Normals - points after octree stage



(e) RGB - points after kd-tree stage *(f)* Normals - points after kd-tree stage

Figure 6.10: Visualization of raw point clouds and point clouds after octree and kd-tree stages

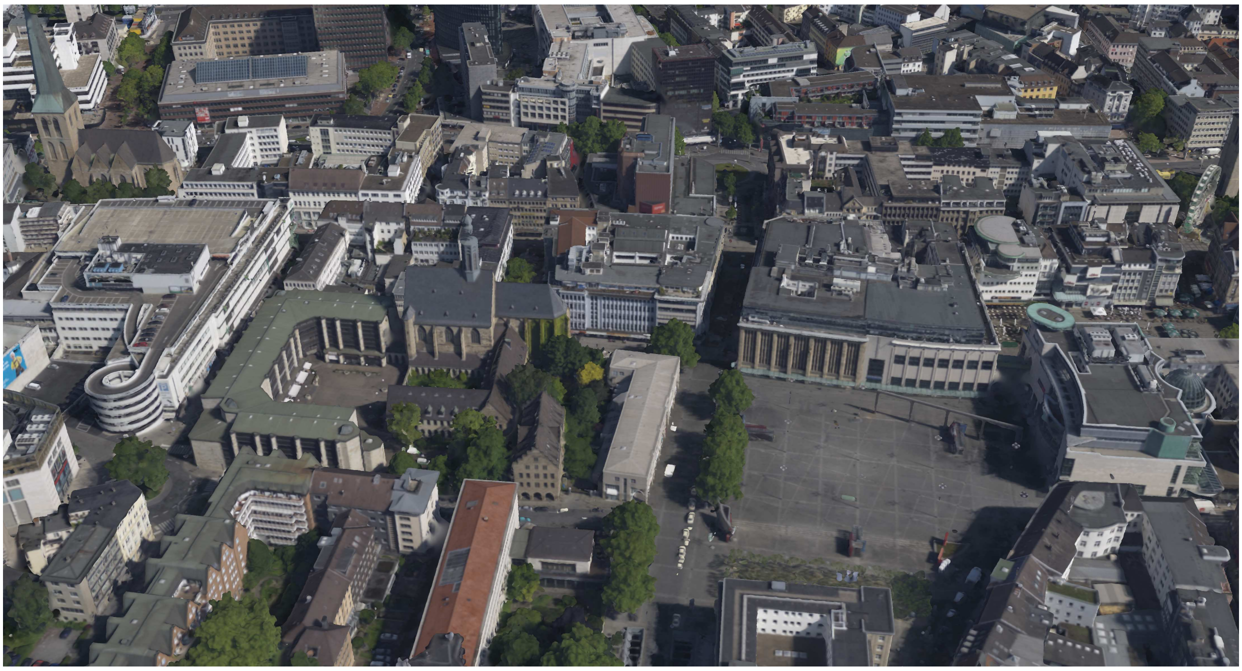


Figure 6.11: Textured mesh generated by the the proposed approach for the Dortmund dataset (14cm avg. GSD). Note that no effects of tiled processing are observable.

which is important for further mesh refinement algorithms as for example proposed in [Kobbelt et al., 1998]. Figure 6.13(c)-(f) depicts more detailed views of a sub area containing stairs. Thereby reconstructions are based on four different processing scenarios. Figure 6.13(d) shows the results for the 2.5D gridding approach based on all, not-filtered, points derived by the proposed multi-baseline approach. Figure 6.13(c) depicts results of the Poisson reconstruction for the same set of input points. For this sample area Poisson reconstruction delivers little smoother results while the degree of detail is comparable. In comparison to the 2.5D surface no effects resulting from real 3D structure as for example the handrails are introduced. The meshes depicted in figure 6.13(e)(f) are based on the points derived by the proposed fusion module. While, comparable to each other, the degree of detail is less then for the scenarios using all non-filtered points. However, despite the marginally reduced precision, extracted geometry is topological correct and the processing time can be reduced from 10.15 hours to 5.65 hours (including fusion and meshing). Note that the degree of details in the actual fused clouds might be larger since within both post-meshing steps additional smoothing is applied. The processing times could be further improved by further parallelization, since large parts of the algorithm are running in a serial section. To demonstrate the quality of extracted meshes, figure 6.14 shows examples of reconstructed surfaces. As typical for Poisson reconstruction melting effects are observed in particular abrupt changes of geometry in combination with sparse input data. Beside facades details as windows, small 3D structures, for example pillars and thin architectural elements are correctly mapped (examples marked by green rectangles). This is only problematic for fine structures which are matched with only limited redundancy (example marked by red rectangle). Except of few areas blunder-free and topological correct meshes are obtained which could be further improved in terms of precision within a subsequent refinement step.

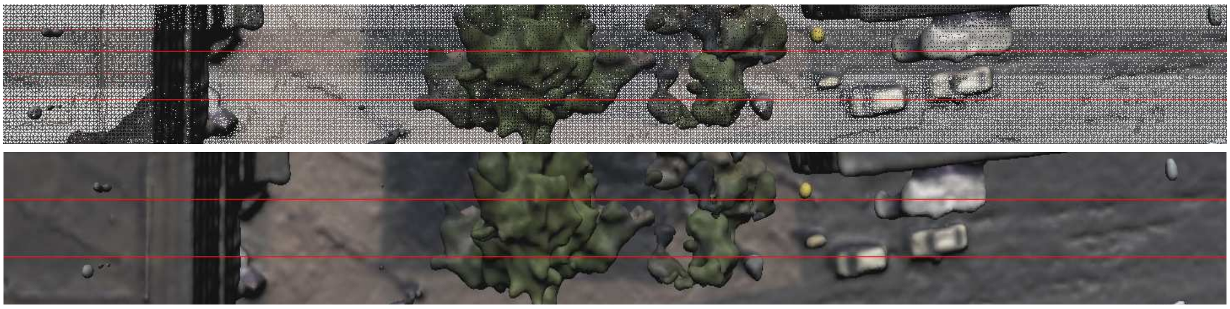
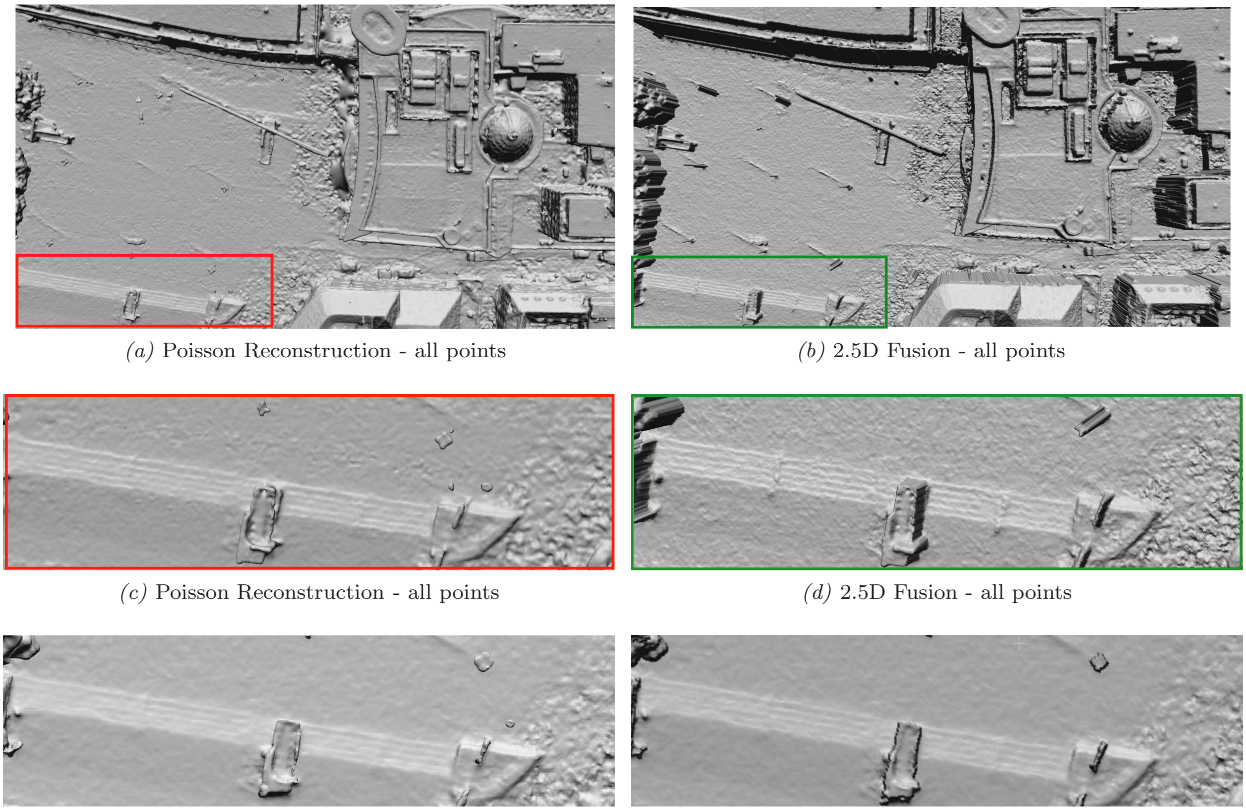


Figure 6.12: Upper: Triangle meshes of two overlapping, neighboring tiles.



(a) Poisson Reconstruction - all points

(b) 2.5D Fusion - all points

(c) Poisson Reconstruction - all points

(d) 2.5D Fusion - all points

(e) Poisson Reconstruction - fused points

(f) 2.5D Fusion - fused points

Figure 6.13: Visualization of meshes derived by Poisson reconstruction and our 2D meshing for different set of input points.

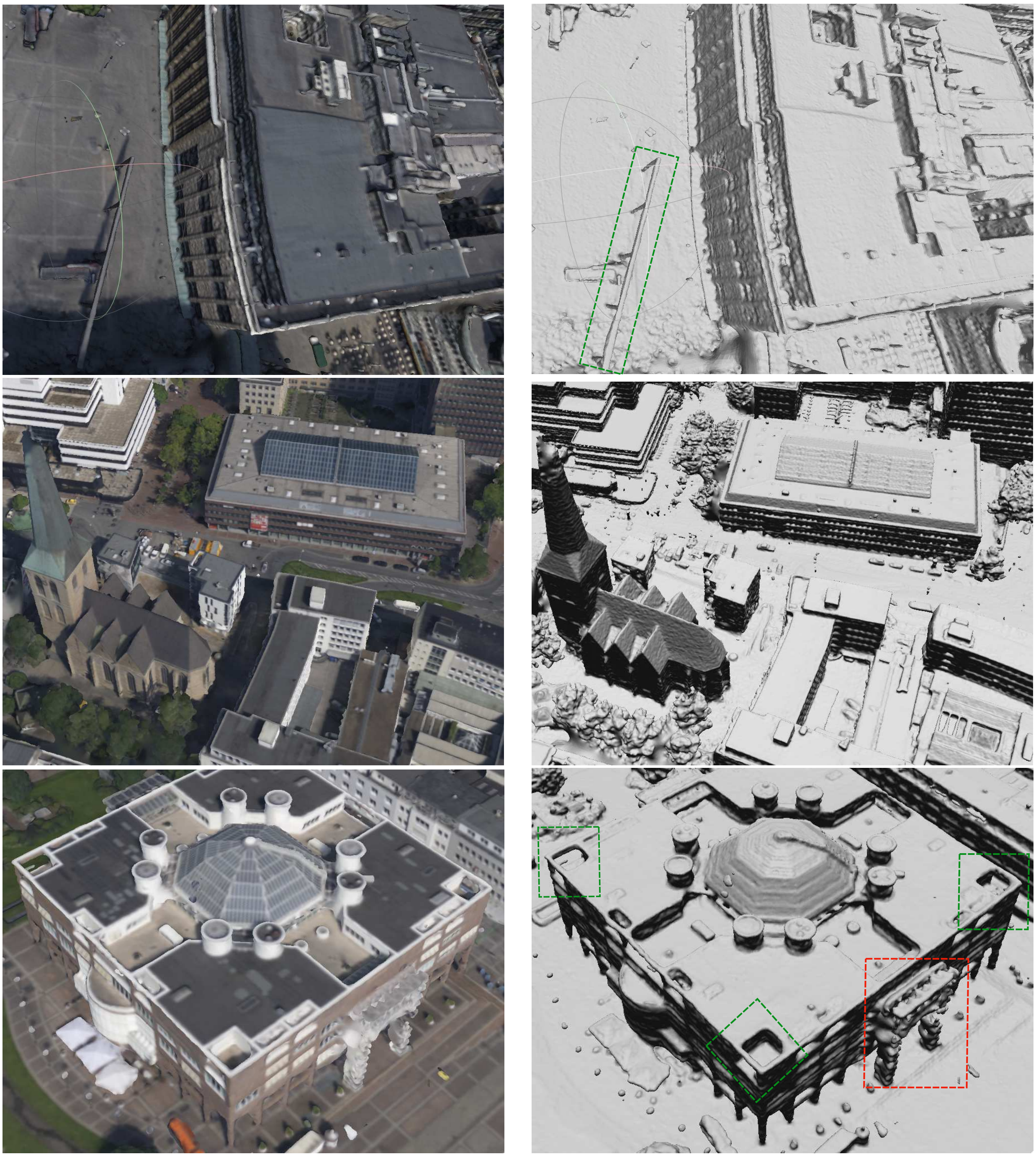


Figure 6.14: Meshes generated from the Dortmund oblique data set. Facade structure, architectural details (green rectangles) and surfaces at undercutting structure can be reconstructed. Thin structures only sparsely represented in point clouds lead to limited correctness of topology (red rectangle).

Chapter 7

Summary and Outlook

In this work we presented an image-based reconstruction pipeline producing point clouds, elevation data and meshes for 2.5D and 3D scenes. In the following the proposed core algorithms and findings are summarized, then limitations and future work will be highlighted.

7.1 Summary

Image Rectification We tested three methods for the process of image rectification. The most common one is based on homographic mapping on virtual image planes. A second, more advanced derives homograpies minimizing projective distortions. We evaluated the performance based on Strecha's benchmark and verified that latter, as expected, delivers slightly more precise results. However, both rectification methods fail on camera configurations of pure forward motion. This problem can be tackled using polar rectification based on half epipolar spaces. Despite within our tests dense matching results utilizing polar rectification are slightly less precise, this algorithm might be most practicable from an implementation point of view since it is capable of rectifying imagery of arbitrary configurations.

Stereo Matching Within the proposed pipeline we realized a coarse-to-fine implementation of the SGM method for stereo matching. Within the conducted tests memory and processing times for typical large frame airborne reconstructions were reduced by more than 60% and 30%. These values were obtained for rather flat terrain and are even more significant for heavily undulating terrain. For another example on a scene possessing an increased field of depth the memory demand and computational load could be reduced by approximately 90%. Furthermore, it was clarified that our hierarchical SGM implementation outerperforms the classical method in areas of weak textures, because limiting disparity search ranges and therefore reducing ambiguities. On the other hand small structures possessing large height differences to its neighborhood might be lost in the course of hierarchical matching. However, in practice theses objects often are mapped in additional views from different directions and structure can be correctly extracted. Although three different matching costs were implemented, the Census correlation turned out to be the most robust regarding different types of scenes and signal-to noise ratios. The Census correlation relies on matching windows, therefore fronto-parallel effects are introduced. This limitation could be overcome by extracting a surface on coarser levels and warping of matching windows according to the scene geometry.

Multi-View Forward Intersection To improve precision and reliability, stereo observations were linked across the single models. Thereby redundant observations are checked for geometric consistency and refined using multi-view forward intersection. We formulated latter tasks for the stereo normal case as a problem minimizing the reprojection error for which a direct solution can be obtained. This avoids matrix inversion in the course of solving a linear system and results in fast processing times. The proposed strategy is fast and capable of removing the gross of blunders contained in the single disparity maps.

2.5D Fusion and Meshing Based on the points generated by the multi-view approach a method for the derivation of digital elevation models was implemented. Generated points are assigned to 2D grids and cell-wise median filtering is performed. We showed that by a simple modification of this approach, much cleaner geometry at depth discontinuities (for example building edges) can be achieved. Moreover, following a tiling scheme guaranteeing scalability to country-wide projects was implemented. Based on a benchmark on DSM generation we verified that our solution is amongst the top performing algorithms regarding processing time and precision. Moreover, we proposed a strategy for mesh generation from DSM data. This involves extraction of geometry adaptive meshes based on RQTs and re-meshing procedures at depth discontinuities to account for inaccurate normals. The latter simplifies subsequent processing steps as mesh texturing for visualization and possible future classification tasks. Furthermore, the proposed mesh extraction guarantees matching (crack-free) triangulations of unlimited scalability.

3D Fusion and Meshing To cope with data sets capturing real 3D geometry a second fusion strategy was proposed. Due to the satisfying results of 2.5D fusion approach we base the 3D fusion on median filtering as well. A method for the extraction of robust normals using modified RQTs for mesh extraction in image space was introduced. Based on these depth discontinuity preserving triangulations, normals for each depth can be extracted. Single observations and their attributes are sorted to an multi-scale linear octree filtering less precise observations. Then, locally proximate observations are identified by cylinder-based queries and subject to median filtering in direction of point normals. In oder to robustify results the filtering procedure is carried out iteratively, but using KD-tree structures to reduce processing times. From the resulting oriented points a surface can be extracted using the Poisson reconstruction. We evaluated our algorithm utilizing the Middlebury multi-view benchmark leading to results comparable to the state-of the art MVS algorithms. Tests on the Fountain benchmark verified improved precisions of depths. Scalability can be achieved by employing tile-wise processing. Employing an oblique airborne dataset the capability of extracting topological correct meshes featuring decent precision and geometric consistent tile borders was verified.

7.2 Limitations and Outlook

Given sufficient image similarity across neighboring views, the proposed MVS pipeline is robust regarding type of input imagery and parametrization. This is partly due to forcing geometric consistency across stereo models and due to the fact that optimization forcing spatial smoothness is carried out repeatedly, more precisely the hierarchical Semi Global Matching is carried out for each suitable image pair. However, algorithmic design based on pair-wise matching involves some limitations:

- Performing multiple optimizations for each image, that is matching a base image against a set of match images, of course implies considerable computational load. Although our algorithm is comparably fast, for country-scale projects processing times may amount several months which leaves room for further improvement.
- Our matching method only incorporates stereo image similarity within the data term of the optimization procedure. Incorporating multi-view image similarity may certainly improve results with respect to sub-pixel accuracy.
- Stereo models deliver smooth surfaces possessing specious low noise levels. However, single models might be inconsistent due to errors in interior and exterior orientations and model errors within the stereo approach. This becomes obvious when analyzing the fused models which often posses increased noise and inconsistencies, e.g. multiple layers. Although these effects most often can be eliminated by the proposed median-based filter/fusion techniques, resultant surfaces might be sub-optimal.

The problems mentioned above can be overcome by designing algorithms in a different way representing the scene by object space entities from an early stage (e.g level sets, meshes, etc.). This way optimization incorporating shape priors and multi-image consistency can be performed once per surface entity in object space. Furthermore, foreshortening effects due to rectangular matching windows can be avoided. Being aware of the scene geometry, matching windows or patches can be adapted accordingly. In future work we plan to improve our method by refinement of the extracted 2.5D and 3D meshes. Since topology of the obtained meshes is reliable, methods as [Delaunoy et al., 2008] could be adapted for subsequent mesh refinement. In terms of computational complexity it has to be investigated whether mesh priors generated by our algorithm utilizing reduced resolution pyramid imagery deliver sufficient quality and detail. Furthermore we would like to investigate possibilities to directly extracted meshes from the fused point clouds. Replacement of the Poisson reconstruction could reduce melting effects, additional smoothing and erroneous interpolations in areas of sparse data. Interesting algorithms include [Jancosek and Pajdla, 2011], [Bodenmueller, 2009] and [Bernardini et al., 1999].

Despite crack-free meshes generated from 2.5D elevation maps are scalable without limitations, for 3D meshes the connection between the single mesh tiles has to be investigated. This issue has been tackled by various other works for example [Vu, 2011], [Wuttke et al., 2012]. Especially latter seem to be promising since geometry at borders of the tiles generated by our approach are close to identical.

To this point the proposed algorithm is able to deal with scale variances within data sets using a multi-resolution octree structure in combination with kd-trees. However, so far the scene is separated by point cubes of identical sizes which then are subject to filtering/fusion. Since processing times increase with the depth of octrees and depths are limited due to numerical reasons the algorithm is not yet applicable for arbitrary scale variances. In future we plan to implement data structures managing point tiles of varying dimensions to tackle this issue.

High frame rates of aerial camera systems offer the possibility to acquire data at low flighting altitudes possessing GSDs at cm-level. As a result generated surface meshes offer impressive detail. Furthermore, as clarified in this work, when utilizing oblique imagery facade structure can be reconstructed. While this work is focused on geometric reconstruction in future work we want to investigate semantic reconstruction. While scene classification delivers valuable information for example for DTM production [Pfeifer et al., 2001] and extraction of building models [Verdie et al., 2015], it also could power geometric reconstruction itself.

Chapter 8

Appendix

8.1 Parametric Matching Costs

In this section the formulas of parametric matching costs are given. Let $\mathbf{p}$ denote an image coordinate in the first image $\mathbf{I}$ and $\mathbf{q}$ denote image coordinates in a second view $\mathbf{I}'$. Probably the simplest *parametric matching costs* is the pixel-wise absolute difference AD

$$C_{ad}(\mathbf{p},\mathbf{q}) = |\mathbf{I}(\mathbf{p}) - \mathbf{I}'(\mathbf{q}+\mathbf{p})| \tag{8.1}$$

and the more robust sum of absolute differences (SAD), operating on intensity differences in the local neighborhood (typically rectangular windows)

$$C_{sad}(\mathbf{p},\mathbf{q}) = \sum_{\mathbf{p}\in N_{\mathbf{p}}} |\mathbf{I}(\mathbf{p}) - \mathbf{I}'(\mathbf{q}+\mathbf{p})| \tag{8.2}$$

Paying respect to offsets in radiometry the zero-mean sum of absolute differences (ZSAD) first subtracts the mean intensities $\bar{\mathbf{I}}$ and $\bar{\mathbf{I}}'$ in the respective windows before computing the SAD:

$$C_{zsad}(\mathbf{p},\mathbf{q}) = \sum_{\mathbf{p}\in N_{\mathbf{p}}} |\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}(\mathbf{p}) - \mathbf{I}'(\mathbf{q}+\mathbf{p}) + \bar{\mathbf{I}}'(\mathbf{q}+\mathbf{p})| \tag{8.3}$$

$$\bar{\mathbf{I}}(\mathbf{p}) = \frac{1}{N} \sum_{\mathbf{p}\in N_{\mathbf{p}}} \mathbf{I}(\mathbf{p}). \tag{8.4}$$

Analogously matching cost based on the sum of squared differences (SSD and ZSSD) can be computed, which in presence of Gaussian noise are optimal estimators.

$$C_{ssd}(\mathbf{p},\mathbf{q}) = \sum_{\mathbf{p}\in N_{\mathbf{p}}} (\mathbf{I}(\mathbf{p}) - \mathbf{I}'(\mathbf{q}+\mathbf{p}))^2 \tag{8.5}$$

$$C_{zssd}(\mathbf{p},\mathbf{q}) = \sum_{\mathbf{p}\in N_{\mathbf{p}}} (\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}(\mathbf{p}) - \mathbf{I}'(\mathbf{q}+\mathbf{p}) + \bar{\mathbf{I}}'(\mathbf{q}+\mathbf{p}))^2 \tag{8.6}$$

However, SSD is sensitive to outliers which frequently occur for example at depth discontinuities and non-Lambertian surfaces. The expansion of the SSD matching cost leads to

$$C_{ssd}(\mathbf{p},\mathbf{q}) = \sum_{\mathbf{p}\in N_{\mathbf{p}}} \mathbf{I}(\mathbf{p})^2 + \mathbf{I}'(\mathbf{p}+\mathbf{q})^2 - 2\mathbf{I}(\mathbf{p})\mathbf{I}'(\mathbf{q}+\mathbf{p}). \tag{8.7}$$

Thereby the first two terms represent the image energies, the third term defines the cross correlation term. Normalizing the cross correlation term by the image energies leads to the normalized cross correlation (NCC)

$$C_{ncc}(\mathbf{p},\mathbf{q}) = \frac{\sum_{\mathbf{p}\in N_{\mathbf{p}}} \mathbf{I}(\mathbf{p})\mathbf{I}'(\mathbf{q}+\mathbf{p})}{\sqrt{\sum_{\mathbf{p}\in N_{\mathbf{p}}} \mathbf{I}(\mathbf{p})^2 \sum_{\mathbf{p}\in N_{\mathbf{p}}} \mathbf{I}'(\mathbf{p}+\mathbf{q})^2}} \tag{8.8}$$

Analogously the zero mean normalized cross correlation (ZNNC) can be derived as

$$C_{zncc}(\mathbf{p},\mathbf{q}) = \frac{\sum_{\mathbf{p}\in N_{\mathbf{p}}} (\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}(\mathbf{p}))(\mathbf{I}'(\mathbf{q}+\mathbf{p}) - \bar{\mathbf{I}}'(\mathbf{q}+\mathbf{p}))}{\sqrt{\sum_{\mathbf{p}\in N_{\mathbf{p}}} (\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}(\mathbf{p}))^2 \sum_{\mathbf{p}\in N_{\mathbf{p}}} (\mathbf{I}'(\mathbf{p}+\mathbf{q}) - \bar{\mathbf{I}}'(\mathbf{p}+\mathbf{q}))^2}}. \tag{8.9}$$

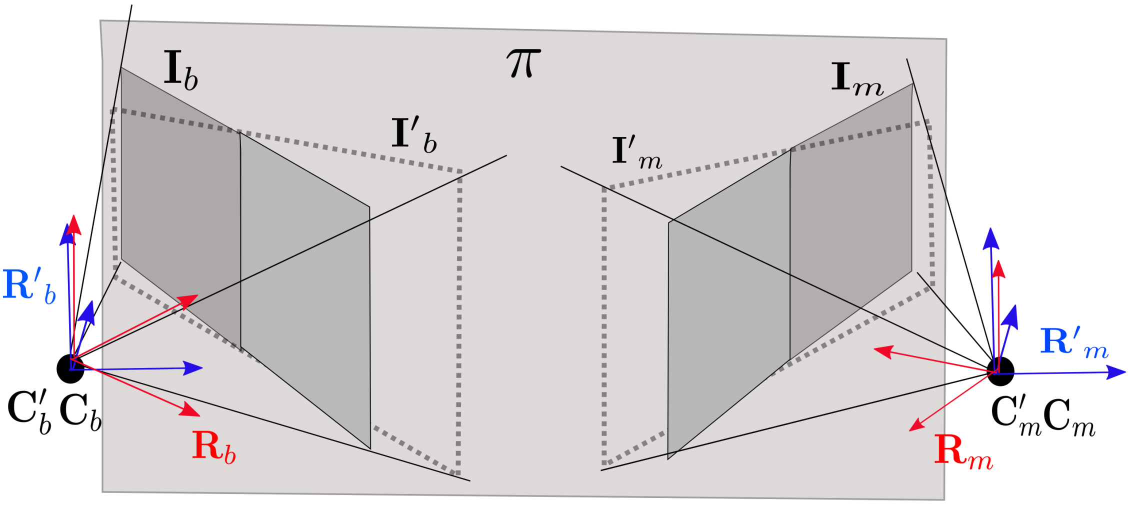


Figure 8.1: Visualization of the rectification method proposed by [Fusiello et al., 2000]. The original views are depicted in dark gray. The bright gray plane π represents the common image plane to which the views are projected to. Note that $\mathbf{R'}_b$ and $\mathbf{R'}_m$ are identical. Rectified images are depicted by dashed lines and are located in the plane π.

8.2 Image Rectification

In this section the three implemented rectification methods are explained. Throughout the following discussion we denote original images by $\mathbf{I}_b$ and $\mathbf{I}_m$ and their rectified versions by $\mathbf{I}_b'$ and $\mathbf{I}_m'$.

8.2.1 Rectification Based on Homographies

Fusiello's Method

Probably the most popular image rectification method was proposed in [Fusiello et al., 2000]. It is based on projecting the original images to two virtual views, both possessing the identical image plane. Moreover, coordinates of rectified and original camera positions are identical (see figure 8.1), therefore

$$\mathbf{C'}_b = \mathbf{C}_b \qquad \mathbf{C'}_m = \mathbf{C}_m. \tag{8.10}$$

The rotations of the virtual views $\mathbf{R'}_b$ and $\mathbf{R'}_m$ are specified such that x-axes of the camera coordinate systems are parallel to translation of the original camera positions

$$\mathbf{R'}_{b,1} = \mathbf{R'}_{m,1} = \frac{\mathbf{C}_b - \mathbf{C}_m}{\|\mathbf{C}_b - \mathbf{C}_m\|_2}. \tag{8.11}$$

Thereby the second lower index i denotes the i-th row of the respective matrix. The y-axis of virtual rotations are required to be orthogonal to the new x-axis and is chosen to

$$\mathbf{R'}_{b,2} = \mathbf{R'}_{m,2} = 0.5(\mathbf{R}_{b,3} + \mathbf{R}_{m,3}) \times \mathbf{R'}_{b,1}. \tag{8.12}$$

The z-axes of rectified views $\mathbf{R'}_{b,3}$ $\mathbf{R'}_{m,3}$ are chosen orthogonal to the new x- and y-axes completing rotations of rectified images. The focal lengths of virtual views are averaged according to

$$f' = f'_m = f'_b = 0.5(f_b + f_m). \tag{8.13}$$

For non-quadratic pixels f_y and f_x are averaged separately. The design strategy for $\mathbf{R'}_b$ and $\mathbf{R'}_m$ assures epipoles at infinity and epipolar lines $\mathbf{l'}_b$ and $\mathbf{l'}_m$ to be parallel and horizontal. Equation 8.13 ensures that object points are mapped to identical image rows across the two rectified views. Using equation 8.13 the camera matrices of the virtual views are given by

$$\mathbf{K'}_b = \mathbf{K'}_m = \begin{bmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{8.14}$$

The remap functions ϕ are defined as

$$\mathbf{x'}_b \ = \phi(\mathbf{x}_b) = \mathbf{H}_b\mathbf{x}_b \quad = \mathbf{K'}_b\mathbf{R'}_b\mathbf{R}_b^\top\mathbf{K}_b^{-1} \ \ \mathbf{x}_b \tag{8.15}$$

$$\mathbf{x'}_m = \phi(\mathbf{x}_m) = \mathbf{H}_m\mathbf{x}_m = \mathbf{K'}_m\mathbf{R'}_m\mathbf{R}_m^\top\mathbf{K}_m^{-1}\mathbf{x}_m. \tag{8.16}$$

In other words the homogeneous vector $\mathbf{x'}$ is derived by first transferring the original vector $\mathbf{x}$ to the world coordinate system and then to the pixel coordinate system defined by the virtual views ($\mathbf{R'}$,$\mathbf{K'}$,$\mathbf{C'}$). So far the pixel coordinate systems of virtual views posses origins with respect to the principal axis. More common are coordinate systems with the origins located at the upper-left image corners. This can be achieved by updating $\mathbf{K'}_b$ and $\mathbf{K'}_m$ utilizing the dimensions of the rectified images. The dimensions can be found by transforming the old corner coordinates using equation 8.16 and 8.15. This leads to x'_{min}, x'_{max}, y'_{min} and y'_{max} for each of the views. The dimensions of rectified images are calculated as

$$\begin{aligned} colums &= ceil(x'_{max}) - floor(x'_{min}) \\ rows &= ceil(y'_{max}) - floor(y'_{min}). \end{aligned} \tag{8.17}$$

Then interior orientations are completed by updating the camera matrices of rectified views according to

$$\mathbf{K'}_b = \begin{bmatrix} f' & 0 & -x'_{b,min} \\ 0 & f' & -y'_{b,min} \\ 0 & 0 & 1 \end{bmatrix} \tag{8.18}$$

$$\mathbf{K'}_m = \begin{bmatrix} f' & 0 & -x'_{m,min} \\ 0 & f' & -y'_{m,min} \\ 0 & 0 & 1 \end{bmatrix}. \tag{8.19}$$

While straight-forward to implement this approach has one significant drawback: For epipoles of two views which are close to zero, that is motion in viewing direction, dimensions of virtual views as well as distortions become huge and the method loses practicality.

Loop’s Method

Another method based on homographies was proposed by [Loop and Zhang, 1999]. Thereby homographies are computed in a way such that perspective distortions in the rectified image pairs are minimized. Let $\mathbf{F'}$ be the fundamental matrix of a rectified image pair:

$$\mathbf{F'} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{8.20}$$

The first property of this fundamental matrix is that the epipoles $\mathbf{e'} = (1, 0, 0)$ of both, base and match images are at infinity which is easily verified by

$$\mathbf{F'e'}_b = \mathbf{0} = \mathbf{F'}^\top\mathbf{e'}_m. \tag{8.21}$$

Furthermore, two homogeneous image coordinates which share the same y component $\mathbf{x'}_b = (x_1, y, 1)$ and $\mathbf{x'}_m = (x_2, y, 1)$ satisfy the epipolar constraint:

$$\mathbf{x'}_b^\top\mathbf{F'x'}_m = 0 \tag{8.22}$$

Formulating this equation with respect to the coordinates of the original images transformed by the homographies $\mathbf{H}_b$ and $\mathbf{H}_m$ leads to:

$$\mathbf{x}_b^\top \mathbf{H}_b^\top \mathbf{F'} \mathbf{H}_m \mathbf{x}_m = \mathbf{x}_b^\top \mathbf{F} \mathbf{x}_m = 0. \tag{8.23}$$

Now the task is to design the homographies $\mathbf{H}_b$ and $\mathbf{H}_m$ in a way such that projective distortions are minimized. Therefore let $\mathbf{H}$ be parametrized as

$$\mathbf{H} = \begin{bmatrix} \mathbf{u}^\top \\ \mathbf{v}^\top \\ \mathbf{w}^\top \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix} \tag{8.24}$$

This can be further composed by splitting the homography into a projective transformation, a similarity transformation and a shearing transformation

$$\mathbf{H} = \mathbf{H}_s \mathbf{H}_r \mathbf{H}_p. \tag{8.25}$$

More precisely the single components are designed as

$$\mathbf{H} = \begin{bmatrix} s_1 & s_2 & s_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_2 - v_3 w_2 & v_3 w_1 - v_1 & 0 \\ v_1 - v_3 w_1 & v_2 - v_3 w_2 & v_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_1 & w_2 & 1 \end{bmatrix}. \tag{8.26}$$

The projective transformation transfers the epipoles in the original images to epipoles at infinity in the rectified images. A point $\mathbf{p}_i = (p_{x,i}, p_{y,i}, 1)$ is mapped by $\mathbf{H}_p$ to $(p_{x,i}/w_i, p_{y,i}/w_i, 1)$ with $w_i = w_1 p_{x,i} + w_2 p_{y,i}$. Identical weights w_i would imply a purely affine transformation. The goal of the approach is to choose w_1 and w_2 in a way such that the projective transform is as affine as possible. Therefore the authors seek to minimize the variance of all weights and the weight of the image center:

$$C = \sum_{i=1}^{n} \left[\tfrac{w_i - w_c}{w_c}\right]^2 = \sum_{i=1}^{n} \left[\tfrac{\mathbf{w}^\top (\mathbf{p}_i - \mathbf{p}_c)}{\mathbf{w}^\top \mathbf{p}_c}\right]^2 = \frac{\mathbf{w}^\top \mathbf{P} \mathbf{P}^\top \mathbf{w}}{\mathbf{w}^\top \mathbf{p}_c \mathbf{p}_c^\top \mathbf{w}} \tag{8.27}$$

with the $3 \times n$ matrix

$$\mathbf{P} = \begin{bmatrix} p_{1,u} - p_{c,u} & p_{2,u} - p_{c,u} & \cdots & p_{n,u} - p_{c,u} \\ p_{1,v} - p_{c,v} & p_{2,u} - p_{c,v} & \cdots & p_{n,v} - p_{c,u} \\ 0 & 0 & \ldots & 0 \end{bmatrix} \tag{8.28}$$

The cost can be computed in similar fashion for both images. The optimal projective transforms fully characterized by $\mathbf{w}_b$ and $\mathbf{w}_m$ can be obtained by minimizing the functional given by summing the costs $C_b + C_m$. However, the vectors $\mathbf{w}_b$ and $\mathbf{w}_m$ are not independent and are related by epipolar geometry. Following [Hartley and Zisserman, 2004] corresponding lines or directions $\mathbf{l}_b$ and $\mathbf{l}_m$ across two views satisfy

$$\mathbf{l}_m = \mathbf{F}[\mathbf{e}_b]_\times \mathbf{l}_b \tag{8.29}$$

where $\mathbf{F}$ is the fundamental matrix and $\mathbf{e}_b$ is the epipole. Let $\mathbf{z}_b = (\alpha, \beta, 0)$ be a direction in the first image $\mathbf{I}_b$. Using the result from equation 8.29 and parameterizing the weight vector as $\mathbf{w}_b = [\mathbf{e}_b]_\times \mathbf{z}$ the correspondence is obtained by $\mathbf{w}_m = \mathbf{F}\mathbf{z}$. Substitution into equation 8.27 leads to the functional

$$C_b + C_m = \frac{\mathbf{z}^\top [\mathbf{e}]_\times^\top \mathbf{P}_b \mathbf{P}_b^\top [\mathbf{e}]_\times \mathbf{z}}{\mathbf{z}^\top [\mathbf{e}]_\times^\top \mathbf{p}_{b,c} \mathbf{p}_{b,c}^\top [\mathbf{e}]_\times \mathbf{z}} + \frac{\mathbf{z}^\top \mathbf{F}^\top \mathbf{P}_m \mathbf{P}_m^\top \mathbf{F} \mathbf{z}}{\mathbf{z}^\top \mathbf{F}^\top \mathbf{p}_{m,c} \mathbf{p}_{m,c}^\top \mathbf{F} \mathbf{z}} := \frac{\mathbf{z}^\top \mathbf{A}_b \mathbf{z}}{\mathbf{z}^\top \mathbf{B}_b \mathbf{z}} + \frac{\mathbf{z}^\top \mathbf{A}_m \mathbf{z}}{\mathbf{z}^\top \mathbf{B}_m \mathbf{z}} \tag{8.30}$$

to be minimized. Using the general homogeneous point coordinates $\mathbf{p}_{x,y} = [x, y, 1]$ and the center coordinates $\mathbf{p}_c = [\frac{w-1}{2}, \frac{h-1}{2}, 1]$ the components in equation 8.30 $\mathbf{P}\mathbf{P}^\top$ and $\mathbf{p}_c \mathbf{p}_c^\top$ can be further simplified to

$$\mathbf{P}\mathbf{P}^\top = \frac{wh}{12} \begin{bmatrix} w^2 - 1 & 0 & 0 \\ 0 & h^2 - 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8.31}$$

and

$$\mathbf{p}_c\mathbf{p}_c^\top = \frac{1}{4}\begin{bmatrix} (w-1)^2 & (w-1)(h-1) & 2(w-1) \\ (w-1)(h-1) & (h-1)^2 & 2(h-1) \\ 2(w-1) & 2(h-1) & 4 \end{bmatrix} \tag{8.32}$$

The functional given by equation 8.30 is a non-linear optimization problem with respect to $\mathbf{z} = [\alpha, \beta]$. Remember that $\mathbf{z}$ is a direction and defined up to a scale factor, thus we can set $\beta = 1$ without loss of generality. The minimum of equation 8.30 is obtained for the first derivative with respect to α equating to zero. Starting with an initial guess derived by minimizing C_b and C_m independently the minimizer of the non-linear functional can be found by iteratively solving for α. This fully specifies the weights $\mathbf{w}_b$ and $\mathbf{w}_b$ and parametrizes the projective transform $\mathbf{H}_P$ transferring epipoles to infinity for the rectified images. In the next step the similarity transform is derived such that epipolar lines in the rectified images are horizontal. The known fundamental matrix $\mathbf{F}$ of original images and the one of the rectified images $\mathbf{F}'$ (equation 8.23) are related by

$$\mathbf{F} = \mathbf{H}_b^\top \mathbf{F}' \mathbf{H}_m \tag{8.33}$$

By comparison of the single terms in this vector equation the parameters v_1, v_2 and v_3 in $\mathbf{H}_{b,r}$ and $\mathbf{H}_{m,r}$ can be eliminated which results in the similarity transform dependent of $\mathbf{w}_b$, $\mathbf{w}_m$ and $v_{m,3}$ solely

$$\mathbf{H}_{b,r} = \begin{bmatrix} F_{32} - w_{b,2}F_{33} & w_{b,1}F_{33} - F_{31} & 0 \\ F_{31} - w_{b,1}F_{33} & F_{32} - w_{b,2}F_{33} & F_{33} + v_{m,3} \\ 0 & 0 & 1 \end{bmatrix} \tag{8.34}$$

$$\mathbf{H}_{m,r} = \begin{bmatrix} w_{m,2}F_{33} - F_{32} & F_{31} - w_{m,1}F_{33} & 0 \\ w_{m,1}F_{33} - F_{31} & w_{m,2}F_{33} - F_{32} & v_{m,3} \\ 0 & 0 & 1 \end{bmatrix}. \tag{8.35}$$

The scalar $v_{m,3}$ can be chosen in a way that the minimum y-pixel coordinates are zero in either of the images. Up to now we specified transformations assuring that epipolar lines are parallel and horizontal. Let $\mathbf{a} = [\frac{w-1}{2}, 0, 1]$, $\mathbf{b} = [w-1, \frac{h-1}{2}, 1]$, $\mathbf{c} = [\frac{w-1}{2}, h-1, 1]$ and $\mathbf{d} = [0\frac{h-1}{2}, 1]$. Setting $s_3 = 0$, the missing parameters s_1 and s_2 of shearing transform can be analytically computed by claiming that the two lines $\mathbf{b} - \mathbf{d}$ and $\mathbf{c} - \mathbf{a}$ are perpendicular and their aspect ratio is preserved. The final homographies are obtained by multiplication of all sub-transforms $\mathbf{H}_b = \mathbf{H}_{b,s}\mathbf{H}_{b,r}\mathbf{H}_{b,p}$ and $\mathbf{H}_m = \mathbf{H}_{m,s}\mathbf{H}_{m,r}\mathbf{H}_{m,p}$. Once the homographies are derived the dimensions of rectified images can be computed in a similar way as within the rectification approach before. Although this rectification method offers a more controlled and interpretable formulation of the rectification process it is still based on homographies and lacks the functionality for pure forward motion configurations.

8.2.2 Polar Rectification

The significant drawback of the incapability of homography-based approaches dealing with general motion can be overcome by rectification using half epipolar lines as proposed by [Pollefeys et al., 1999]. Although the pure forward motion configurations merely occur for traditional nadir data collection, it is more important for imagery acquired by UAVs. Cameras mounted in flight direction, as well as instability of platforms produce images close to the critical configuration. The basic concept of the polar rectification are half-epipolar lines and a re-parametrization of image coordinates from Cartesian coordinates to polar coordinates with the origin located in the epipoles. In a first step the commonly seen areas in the two views to be rectified are determined. Let $\mathbf{e}$ and $\mathbf{e}'$ be the epipoles in the first and second image respectively. The borders of the commonly observed areas are always bounded by the set of extremal epipolar lines defined by the epipole and the image corners. Dependent on the location of the epipole (see figure 8.2, red sections a-i) the two extremal epipolar lines are transferred to the second view and vice versa using

$$\mathbf{l}_m = \mathbf{H}^{-\top}\mathbf{l} \quad \textit{and} \quad \mathbf{l}_b = \mathbf{H}^\top \mathbf{l}_m \tag{8.36}$$

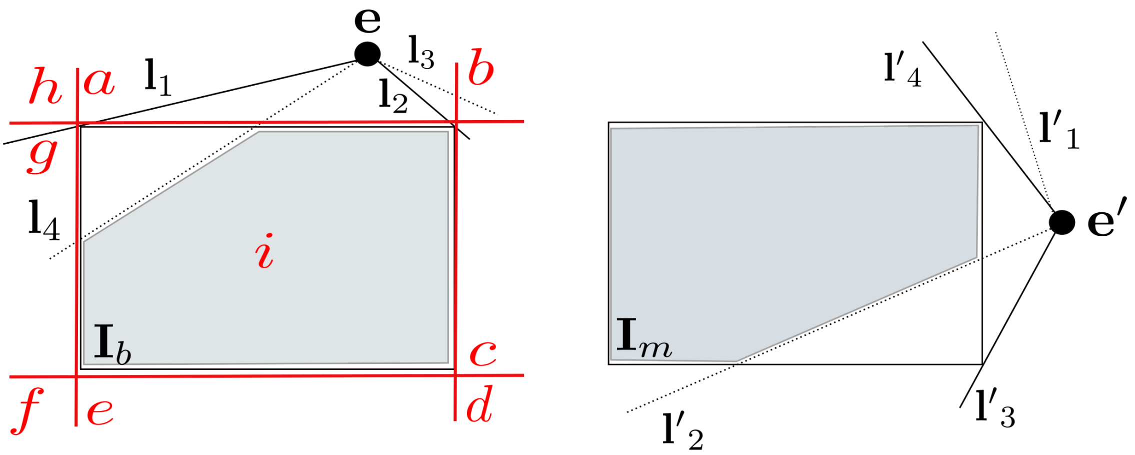


Figure 8.2: left: The common region of the two images $\mathbf{I}_b$ and $\mathbf{I}_m$ to be rectified (depicted in gray) is derived by the extremal epipolar images $\mathbf{l}_{b,1}$, $\mathbf{l}_{b,2}$, $\mathbf{l}_{b,3}$, $\mathbf{l}_{b,4}$ and their correspondences $\mathbf{l}_{m,1}$, $\mathbf{l}_{m,2}$, $\mathbf{l}_{m,3}$, $\mathbf{l}_{m,4}$. Extremal epipolar lines are derived using the position depicted by a-i for $\mathbf{I}_b$ and analogously for $\mathbf{I}_m$.

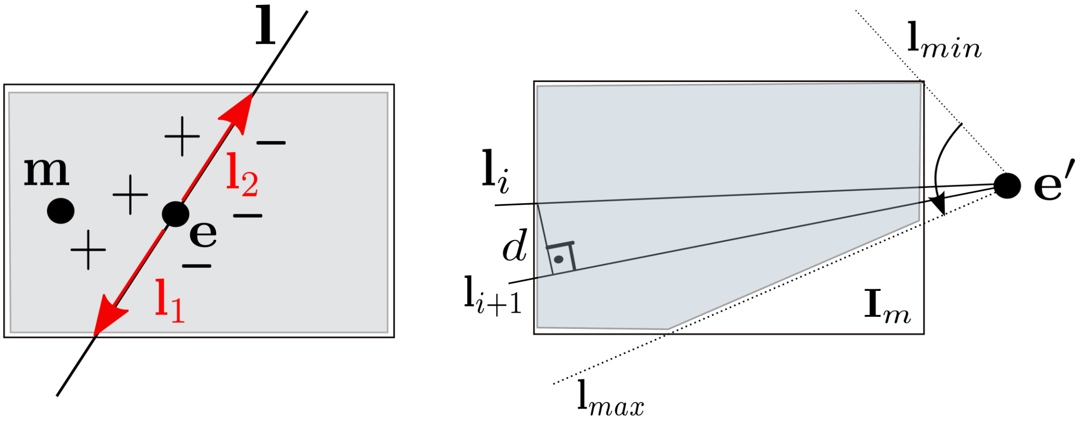


Figure 8.3: Left: Correspondences of half epipolar line $\mathbf{l}_{b,1}$,$\mathbf{l}_{b,2}$ and $\mathbf{l}_{m,1}$,$\mathbf{l}_{m,2}$ are identified by the orientation of the epipolar lines $\mathbf{l}_b$ and $\mathbf{l}_m$. These orientations are derived by a correspondence ($\mathbf{m}_b$,$\mathbf{m}_m$). Indices of denoting the roles of base and match images are not displayed since the approach is similar for both. Right: Images are sampled along $\mathbf{l}_i$ starting from $\mathbf{l}_{min}$ to $\mathbf{l}_{max}$. The angle between consequtive lines is limited by $d = 1$.

with the homography composed of the projection matrices

$$\mathbf{H}^{-\top} = (\mathbf{P}_m^\top)^\dagger \mathbf{P}_b^\top. \tag{8.37}$$

Thereby † denotes the Moore-Penrose pseudo inverse. The transferred extremal epipolar lines define the commonly observed region (figure 8.2, gray areas). The epipoles $\mathbf{e}_b$, $\mathbf{e}_m$ divide each of the two corresponding epipolar lines $\mathbf{l}_b$, $\mathbf{l}_m$ into two line segments, that is $\mathbf{l}_{b,1}$, $\mathbf{l}_{b,2}$, $\mathbf{l}_{m,1}$, $\mathbf{l}_{m,2}$. To identify which of the half epipolar lines correspond across the images the orientations of the full epipolar lines are computed. Therefore, in the original implementation a point correspondence $\mathbf{m}_b = (m_{x,b}, m_{y,b}, 1)$, $\mathbf{m}_m = (m_{x,m}, m_{y,m}, 1)$ is utilized. The orientation of the epipolar line in the first image is given by the sign of $\mathbf{l}_b^\top \mathbf{m}_b$. The orientation of the epipolar line in the second view is then derived by $\mathbf{l}_m^\top \mathbf{m}_m$ (see figure 8.3, left). Based on the two orientations the corresponding half epipolar lines are selected. The authors presume that point correspondences are already available from previous orientation computations which is unfortunately not the case in our pipeline. Therefore we rectify the imagery twice, based on the two possible orientations using low resolution imagery. Then we identify the correct one by analyzing the density of dense matching results. From the correct disparity image the necessary correspondence is derived. Once the correct orientation is known full resolution imagery is re-sampled using polar coordinates. As visualized in figure 8.3, re-sampling is started at the extremal epipolar lines $\mathbf{l}_{min}$. The first line in the rectified base images is defined by sampling along the half epipolar line $\mathbf{l}_{min}$. Similarly the correspondent line in the second view is re-sampled. Subsequently the angle is increased and the next line is resampled. This is repeated until the maximal extremal epipolar line $\mathbf{l}_{max}$ is reached. To avoid loss of information the angle is bounded by the largest distance of consecutive lines to be at at least one pixel as shown in figure 8.3(right), e.g $d < 1pix$. Because angles between consecutive epipolar lines generally vary across the two views, the minimal angle of both tuples is used. The minimal distance and maximal distances along each epipolar line are easily derived by intersections with the image borders. For back transformation to Cartesian coordinates, as needed for structure computation, minimum and maximum distances as well as angular offsets may be stored in a look-up table. Instead we prefer re-computation of these values for back transformation to avoid input and output operations.

8.3 Additional Material for Evaluation of Depth Maps

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,2}(F)$	-9.23	36.87	38.01	63.77	-6.86	15.92	17.33	63.04
$\mathbf{I}_{b,2}(L)$	-8.97	37.08	38.15	60.82	-6.79	15.92	17.31	60.13
$\mathbf{I}_{b,2}(P)$	-7.83	38.54	39.32	56.98	-5.58	17.23	18.11	56.39
$\mathbf{I}_{b,3}(F)$	-9.21	28.71	30.15	67.82	-7.77	12.66	14.85	67.15
$\mathbf{I}_{b,3}(L)$	-9.05	32.29	33.54	63.89	-7.83	12.91	15.10	63.38
$\mathbf{I}_{b,3}(P)$	-8.61	31.74	32.89	52.87	-7.41	12.90	14.87	52.49
$\mathbf{I}_{b,4}(F)$	-8.99	31.94	33.18	67.92	-8.33	10.94	13.75	67.48
$\mathbf{I}_{b,4}(L)$	-8.88	32.55	33.74	64.90	-8.17	10.70	13.46	64.50
$\mathbf{I}_{b,4}(P)$	-7.96	34.03	34.95	58.02	-7.33	11.42	13.57	57.66
$\mathbf{I}_{b,5}(F)$	-6.95	30.07	30.87	66.53	-6.37	10.42	12.21	66.12
$\mathbf{I}_{b,5}(L)$	-6.93	25.70	26.62	64.60	-6.08	9.86	11.58	64.15
$\mathbf{I}_{b,5}(P)$	-7.39	30.19	31.08	60.44	-6.70	10.51	12.46	60.06
$\mathbf{I}_{b,6}(F)$	-6.66	23.45	24.37	60.52	-5.82	9.10	10.80	60.08
$\mathbf{I}_{b,6}(L)$	-6.36	23.60	24.44	59.21	-5.60	8.87	10.49	58.80
$\mathbf{I}_{b,6}(P)$	-5.17	24.39	24.93	59.32	-4.22	9.31	10.22	58.88
$\mathbf{I}_{b,7}(F)$	-3.87	29.57	29.82	56.98	-3.43	9.40	10.01	56.74
$\mathbf{I}_{b,7}(L)$	-4.06	30.54	30.80	56.47	-3.63	9.08	9.78	56.21
$\mathbf{I}_{b,7}(P)$	-4.70	26.06	26.48	43.56	-3.82	9.09	9.86	43.28
$\mathbf{I}_{b,8}(F)$	-6.58	27.63	28.40	45.09	-5.27	9.81	11.14	44.78
$\mathbf{I}_{b,8}(L)$	-6.64	29.01	29.76	47.14	-5.39	9.50	10.93	46.86
$\mathbf{I}_{b,8}(P)$	-6.32	38.71	39.22	43.83	-4.55	12.24	13.06	43.55

Table 8.1: Results for multi-baseline matching, all values expressed in millimeters except of $\%_{val}$ and $\%_{val,3}$. Each base images was matched with 4 neighbors using 3 types of rectifications. Rectification algorithms are [Fusiello et al., 2000] (F), [Loop and Zhang, 1999] (L) and [Pollefeys et al., 1999](P).

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,3}(M1)$	-13.56	124.41	125.14	75.38	-11.02	34.99	36.68	74.54
$\mathbf{I}_{b,3}(M2)$	-8.72	36.99	38.00	65.16	-7.74	13.73	15.76	64.67
$\mathbf{I}_{b,3}(M3)$	-7.71	20.88	22.26	57.17	-6.56	9.85	11.83	56.52
$\mathbf{I}_{b,3}(M4)$	-6.15	14.11	15.39	45.95	-5.61	8.15	9.89	45.67
$\mathbf{I}_{b,3}(M5)$	-5.44	11.20	12.45	36.07	-5.03	7.30	8.87	35.84
$\mathbf{I}_{b,4}(M1)$	-7.88	109.95	110.23	73.69	-8.81	27.94	29.30	72.70
$\mathbf{I}_{b,4}(M2)$	-7.62	33.91	34.76	64.89	-7.05	10.91	12.99	64.49
$\mathbf{I}_{b,4}(M3)$	-6.77	19.87	20.99	58.14	-6.34	8.50	10.60	57.86
$\mathbf{I}_{b,4}(M4)$	-5.99	13.23	14.52	47.67	-5.65	7.33	9.26	47.42
$\mathbf{I}_{b,4}(M5)$	-5.09	10.00	11.22	36.23	-4.81	6.45	8.05	36.02
$\mathbf{I}_{b,5}(M1)$	-4.99	88.18	88.32	70.81	-8.17	24.51	25.84	69.94
$\mathbf{I}_{b,5}(M2)$	-6.93	32.93	33.65	62.11	-6.53	10.44	12.32	61.76
$\mathbf{I}_{b,5}(M3)$	-6.24	18.41	19.44	55.84	-5.98	7.86	9.87	55.59
$\mathbf{I}_{b,5}(M4)$	-5.64	11.22	12.56	44.36	-5.42	6.69	8.61	44.11
$\mathbf{I}_{b,5}(M5)$	-4.75	7.71	9.05	31.19	-4.49	5.47	7.07	30.85
$\mathbf{I}_{b,6}(M1)$	-6.09	87.31	87.53	62.76	-8.27	24.55	25.90	61.95
$\mathbf{I}_{b,6}(M2)$	-6.74	27.39	28.21	53.20	-6.01	9.66	11.38	52.83
$\mathbf{I}_{b,6}(M3)$	-5.86	16.01	17.05	46.47	-5.51	7.09	8.98	46.19
$\mathbf{I}_{b,6}(M4)$	-5.33	10.25	11.55	35.08	-5.03	5.95	7.80	34.82
$\mathbf{I}_{b,6}(M5)$	-5.96	8.62	10.48	13.51	-5.59	5.45	7.81	13.37
$\mathbf{I}_{b,7}(M1)$	-5.20	87.94	88.10	64.75	-5.66	24.89	25.52	64.14
$\mathbf{I}_{b,7}(M2)$	-4.21	31.24	31.52	53.39	-3.89	9.51	10.28	53.15
$\mathbf{I}_{b,7}(M3)$	-3.95	15.91	16.39	45.33	-3.69	6.90	7.82	45.12
$\mathbf{I}_{b,7}(M4)$	-5.20	8.92	10.32	28.32	-4.81	5.64	7.41	28.06
$\mathbf{I}_{b,7}(M5)$	-5.69	6.88	8.93	14.80	-5.31	4.30	6.83	14.60

Table 8.2: Results for multi-baseline matching, all values expressed in millimeters except of $\%_{val}$ and $\%_{val,3}$. Each base images was matched with 6 neighbors using different thresholds of minimal geometric consistent observations $t_{min} = [1...5]$ (M1-M5). Image space accuracy was set to $\sigma_I = 2$.

8.4 Additional Material for Evaluation Fuion of Depth Maps

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,3}(P)$	-0.0062	0.0106	0.0123	70.52	-0.0058	0.0089	0.0106	68.68
$\mathbf{I}_{b,4}(F)$	-0.0060	0.0094	0.0112	70.59	-0.0056	0.0078	0.0096	68.79
$\mathbf{I}_{b,5}(F)$	-0.0055	0.0087	0.0103	70.39	-0.0050	0.0069	0.0085	68.43
$\mathbf{I}_{b,6}(F)$	-0.0055	0.0085	0.0101	70.28	-0.0050	0.0066	0.0083	68.15
$\mathbf{I}_{b,7}(P)$	-0.0056	0.0086	0.0102	68.21	-0.0050	0.0065	0.0082	66.00
$\mathbf{I}_{b,8}(P)$	-0.0062	0.0090	0.0109	60.62	-0.0055	0.0069	0.0088	58.42

Table 8.3: Results for fusion, all values expressed in gsd[pix] except of $\%_{val}$ and $\%_{val,3}$. Each base image was matched with 4 neighbors, fused points were projected to ground truth images.

	δ	σ	δ_{rms}	$\%_{val}$	δ_3	σ_3	$\delta_{rms,3}$	$\%_{val,3}$
$\mathbf{I}_{b,3}(P)$	-0.0072	0.0106	0.0128	68.63	-0.0065	0.0091	0.0112	66.67
$\mathbf{I}_{b,4}(F)$	-0.0080	0.0096	0.0125	69.15	-0.0074	0.0082	0.0111	67.17
$\mathbf{I}_{b,5}(F)$	-0.0061	0.0091	0.0110	69.54	-0.0056	0.0076	0.0095	67.67
$\mathbf{I}_{b,6}(F)$	-0.0056	0.0084	0.0101	66.40	-0.0051	0.0067	0.0084	64.50
$\mathbf{I}_{b,7}(P)$	-0.0033	0.0085	0.0091	60.77	-0.0028	0.0069	0.0075	59.34
$\mathbf{I}_{b,8}(P)$	-0.0051	0.0088	0.0102	47.73	-0.0045	0.0069	0.0082	46.26

Table 8.4: Results for fusion, all values expressed in gsd[pix] except of $\%_{val}$ and $\%_{val,3}$. Each base image was matched with 4 neighbors, fused points were projected to ground truth images.

Bibliography

[Abraham and Förstner, 2005] Abraham, S. and Förstner, W. (2005). Fish-eye-stereo calibration and epipolar rectification. *ISPRS Journal of photogrammetry and remote sensing*, 59(5):278–288.

[Agarwal et al., 2010] Agarwal, S., Snavely, N., Seitz, S. M., and Szeliski, R. (2010). Bundle adjustment in the large. In *Proccedings of the European Conference on Computer Vision (ECCV) 2010*, pages 29–42. Springer.

[Agarwal et al., 2009] Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. (2009). Building rome in a day. In *International Conference on Computer Vision (ICCV) 2009*, pages 72–79. IEEE.

[Amenta et al., 1998] Amenta, N., Bern, M., and Eppstein, D. (1998). The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing*, 60(2):125–135.

[Ayache and Hansen, 1988] Ayache, N. and Hansen, C. (1988). Rectification of images for binocular and trinocular stereovision. In *International Conference on Pattern Recognition (ICPR) 1988*, pages 11–16. IEEE.

[Baker and Binford, 1981] Baker, H. H. and Binford, T. O. (1981). Depth from edge and intensity based stereo. In *Proccedings of the International Joint Conference on Artificial Intelligence (IJCAI) 1981*, pages 631–636, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Baltsavias, 1991] Baltsavias, E. P. (1991). *Multiphoto geometrically constrained matching*. PhD thesis, Diss. Techn. Wiss. ETH Zürich, Nr. 9561.

[Banz et al., 2010] Banz, C., Hesselbarth, S., Flatt, H., Blume, H., and Pirsch, P. (2010). Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In *Proceedings of the International Conference on Embedded Computer Systems (SAMOS) 2010*, pages 93–101. IEEE.

[Belhumeur, 1996] Belhumeur, P. N. (1996). A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260.

[Bernardini et al., 1999] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359.

[Bethmann and Luhmann, 2015] Bethmann, F. and Luhmann, T. (2015). Semi-global matching in object space. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:23–30.

[Bodenmueller, 2009] Bodenmueller, T. (2009). *Streaming surface reconstruction from real time 3D-measurements.* PhD thesis, Technical University Munich.

[Botsch et al., 2002] Botsch, M., Steinberg, S., Bischoff, S., and Kobbelt, L. (2002). Openmesh-a generic and efficient polygon mesh data structure.

[Boykov and Kolmogorov, 2004] Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

[Boykov et al., 1998] Boykov, Y., Veksler, O., and Zabih, R. (1998). Markov random fields with efficient approximations. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 1998*, pages 648–655. IEEE.

[Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

[Cigolle et al., 2014] Cigolle, Z. H., Donow, S., Evangelakos, D., Mara, M., McGuire, M., and Meyer, Q. (2014). A survey of efficient representations for independent unit vectors. *Journal of Computer Graphics Techniques (JCGT)*, 3(2).

[Cormen et al., 2001] Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition.

[Cox et al., 1996] Cox, I. J., Hingorani, S. L., Rao, S. B., and Maggs, B. M. (1996). A maximum likelihood stereo algorithm. *Computer vision and image understanding*, 63(3):542–567.

[Curless and Levoy, 1996] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH) 1996*, pages 303–312. ACM.

[Delaunoy et al., 2008] Delaunoy, A., Prados, E., Piracés, P. G. I., Pons, J.-P., and Sturm, P. (2008). Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *Proceedings of the British Machine Vision Conference (BMVC) 2008*, pages 1–10. BMVA.

[Drouin et al., 2005] Drouin, M.-A., Trudeau, M., and Roy, S. (2005). Geo-consistency for wide multi-camera stereo. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2005*, pages 351–358. IEEE.

[Duane, 1971] Duane, C. B. (1971). Close-range camera calibration. *ISPRS Journal of photogrammetry and remote sensing*, 37:855–866.

[Egnal and Wildes, 2002] Egnal, G. and Wildes, R. (2002). Detecting binocular half-occlusions: empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1127–1133.

[Ernst and Hirschmüller, 2008] Ernst, I. and Hirschmüller, H. (2008). Mutual information based semi-global stereo matching on the gpu. In *Proceedings of the 4th International Symposium on Advances in Visual Computing*, pages 228–239. Springer.

[Faugeras and Keriven, 1998] Faugeras, O. and Keriven, R. (1998). Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *Transactions on Imgage Processing*, 7(3):336–344.

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient belief propagation for early vision. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2004*, pages 261–268. IEEE.

[Freeman and Pasztor, 1999] Freeman, W. and Pasztor, E. (1999). Learning low-level vision. In *Proceedings of the International Conference on Computer Vision (ICCV) 1999*, pages 1182–1189 vol.2. IEEE.

[Fuhrmann and Goesele, 2011] Fuhrmann, S. and Goesele, M. (2011). Fusion of depth maps with multiple scales. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 148:1–148:8, New York, NY, USA. ACM.

[Furukawa et al., 2010] Furukawa, Y., Curless, B., Seitz, S., and Szeliski, R. (2010). Towards internet-scale multi-view stereo. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2010*, pages 1434–1441. IEEE.

[Furukawa and Ponce, 2010] Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376.

[Fusiello et al., 2000] Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22.

[Gargantini, 1982] Gargantini, I. (1982). An effective way to represent quadtrees. *Communications of the ACM*, 25(12):905–910.

[Garland and Heckbert, 1997] Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) 1997*, SIGGRAPH '97, pages 209–216, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Gehrig et al., 2009] Gehrig, S. K., Eberli, F., and Meyer, T. (2009). A real-time low-power stereo vision engine using semi-global matching. In *Computer Vision Systems*, pages 134–143. Springer.

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proccedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2012*, pages 3354–3361. IEEE.

[Geiger et al., 1995] Geiger, D., Ladendorf, B., and Yuille, A. (1995). Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3):211–226.

[Gibson and Marques, 2008] Gibson, J. and Marques, O. (2008). Stereo depth with a unified architecture gpu. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2008*, pages 1–6. IEEE.

[Goesele et al., 2006] Goesele, M., Curless, B., and Seitz, S. (2006). Multi-view stereo revisited. In *Proccedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2006*, volume 2, pages 2402–2409. IEEE.

[Goesele et al., 2007] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections.

[Goldberg and Rao, 1997] Goldberg, A. V. and Rao, S. (1997). Length functions for flow computations. Technical report, NEC Research Institute.

[Gruen, 1985] Gruen, A. (1985). Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187.

[Gruen and Baltsavias, 1988] Gruen, A. W. and Baltsavias, E. P. (1988). Geometrically constrained multiphoto matching. *Photogrammetric Engineering and Remote Sensing*, 54:633–641.

[Grun and Zhang, 2002] Grun, A. and Zhang, L. (2002). Automatic dtm generation from three-line-scanner (tls) images. *ISPRS International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):131–137.

[Haala, 2013a] Haala, N. (2013a). Dense image matching final report. *EuroSDR Publication Series, Official Publication No. 64*, pages 115–145.

[Haala, 2013b] Haala, N. (2013b). The landscape of dense image matching algorithms. *Proceedings of the Photogrammetric Week 2013*, pages 271–284.

[Hancock and Kittler, 1990] Hancock, E. R. and Kittler, J. (1990). Discrete relaxation. *Pattern Recognition*, 23(7):711–733.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[Hermann and Klette, 2012a] Hermann, S. and Klette, R. (2012a). Evaluation of a new coarse-to-fine strategy for fast semi-global stereo matching. *Advances in Image and Video Technology*, pages 395–406.

[Hermann and Klette, 2012b] Hermann, S. and Klette, R. (2012b). Iterative semi-global matching for robust driver assistance systems. In *Asian Conference on Computer Vision Computer Vision (ACCV) 2012*, pages 465–478. Springer.

[Hiep et al., 2009] Hiep, V. H., Keriven, R., Labatut, P., and Pons, J.-P. (2009). Towards high-resolution large-scale multi-view stereo. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2009*, pages 1430–1437. IEEE.

[Hirschmüller, 2008] Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:328–341.

[Hirschmüller et al., 2012] Hirschmüller, H., Buder, M., and Ernst, I. (2012). Memory efficient semi-global matching. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:371–376.

[Hirschmüller and Scharstein, 2007] Hirschmüller, H. and Scharstein, D. (2007). Evaluation of cost functions for stereo matching. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2007*, pages 1 –8. IEEE.

[Irschara et al., 2012] Irschara, A., Rumpler, M., Meixner, P., Pock, T., and Bischof, H. (2012). Efficient and globally optimal multi view dense matching for aerial images. *ISPRS annals of photogrammetry, remote sensing and spatial information sciences*, 1:227–232.

[Ishikawa, 2003] Ishikawa, H. (2003). Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336.

[Jancosek and Pajdla,] Jancosek, M. and Pajdla, T. Cmpmvs - multi-view reconstruction software. http://ptak.felk.cvut.cz/sfmservice. Accessed: 2016-02-3.

[Jancosek and Pajdla, 2011] Jancosek, M. and Pajdla, T. (2011). Multi-view reconstruction preserving weakly-supported surfaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2011*, pages 3121–3128. IEEE.

[Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pages 61–70. Eurographics Association.

[Kazhdan and Hoppe, 2013] Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):29:1–29:13.

[Kobbelt et al., 1998] Kobbelt, L., Campagna, S., Vorsatz, J., and Seidel, H.-P. (1998). Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) 1998*, pages 105–114. ACM.

[Koch et al., 1998] Koch, R., Pollefeys, M., and Gool, L. J. V. (1998). Multi viewpoint stereo from uncalibrated video sequences. In *Proceedings of the 5th European Conference on Computer Vision (ECCV) 1998*, pages 55–71. Springer-Verlag.

[Kolmogorov and Zabih, 2001] Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions via graph cuts. Technical report, Cornell University, Ithaca, NY, USA.

[Kraus, 1994] Kraus, K. (1994). *Photogrammetrie - Band 1.* Ferd. Dümmlers Verlag, ISBN: 3-427-78645-5.

[Krzystek, 1991] Krzystek, P. (1991). Fully automatic measurement of digital elevation models. In *Proceedings of the 43rd Photogrammetric Week*, pages 203–214.

[Kuhn et al., 2014] Kuhn, A., Mayer, H., Hirschmuller, H., and Scharstein, D. (2014). A tv prior for high-quality local multi-view stereo reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV) 2014*, volume 1, pages 65–72. IEEE.

[Kutulakos and Seitz, 1998] Kutulakos, K. N. and Seitz, S. M. (1998). What do photographs tell us about 3d shape? Technical report, Technical Report TR692, Computer Science Dept., U. Rochester.

[Labatut et al., 2007] Labatut, P., Pons, J.-P., and Keriven, R. (2007). Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *Proceedings of the International Conference on Computer Vision (ICCV) 2007*, pages 1–8. IEEE.

[Labatut et al., 2009] Labatut, P., Pons, J.-P., and Keriven, R. (2009). Robust and efficient surface reconstruction from range data. In *Computer Graphics Forum*, volume 28, pages 2275–2290. Wiley Online Library.

[Loop and Zhang, 1999] Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 1999*, volume 1, pages 637–663. IEEE.

[Lorensen and Cline, 1987] Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH) 1987*, SIGGRAPH '87, pages 163–169, New York, NY, USA. ACM.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV) 1999*, volume 2, pages 1150–1157. IEEE.

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 1981*, IJCAI'81, pages 674–679.

[Maas, 1996] Maas, H.-G. (1996). Automatic dem generation by multi-image feature based matching. *International Archives of Photogrammetry and Remote Sensing*, 31:484–489.

[Merrell et al., 2007] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nistér, D., and Pollefeys, M. (2007). Real-time visibility-based fusion of depth maps. In *Proceedings of the International Conference on Computer Vision (ICCV) 2007*, pages 1–8. IEEE.

[Meyer et al., 2010] Meyer, Q., Süßmuth, J., Sußner, G., Stamminger, M., and Greiner, G. (2010). On floating-point normal vectors. In *Computer Graphics Forum*, volume 29, pages 1405–1409. Wiley Online Library.

[Morel and Yu, 2009] Morel, J.-M. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469.

[Newton, 1999] Newton, I. (1999). A method for the automated production of digital terrain models using a combination of feature points, grid points, and filling back points. *Photogrammetric Engineering and Remote Sensing*, 65:713–719.

[Ohta and Kanade, 1985] Ohta, Y. and Kanade, T. (1985). Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154.

[Okutomi and Kanade, 1993] Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.

[OpenMP Architecture Review Board, 2012] OpenMP Architecture Review Board (2012). OpenMP application program interface version 3.1.

[Osher and Sethian, 1988] Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics.*, 79(1):12–49.

[Pajarola, 1998] Pajarola, R. (1998). Large scale terrain visualization using the restricted quadtree triangulation. In *Proceedings of the conference on Visualization 1998*, pages 19–26. IEEE.

[Pajarola, 2002] Pajarola, R. (2002). *Overview of quadtree-based terrain triangulation and visualization*. Department of Information & Computer Science, University of California, Irvine.

[Pajarola et al., 2002] Pajarola, R., Antonijuan, M., and Lario, R. (2002). Quadtin: Quadtree based triangulated irregular networks. In *Proceedings of the conference on Visualization 2002*, pages 395–402. IEEE.

[Pfeifer et al., 2001] Pfeifer, N., Stadler, P., and Briese, C. (2001). Derivation of digital terrain models in the scop++ environment. *Proceedings of OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Terrain Models*, 3612.

[Pierrot-Deseilligny and Paparoditis, 2006] Pierrot-Deseilligny, M. and Paparoditis, N. (2006). A multiresolution and optimization-based image matching approach: An application to surface reconstruction from spot5-hrs stereo imagery. In *Proceedings of the ISPRS Conference Topographic Mapping From Space (With Special Emphasis on Small Satellites)*. ISPRS.

[Pock et al., 2010] Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2010). Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3(4):1122–1145.

[Pollefeys et al., 1999] Pollefeys, M., Koch, R., and Van Gool, L. (1999). A simple and efficient rectification method for general motion. In *Proceedings of the International Conference on Computer Vision (ICCV) 1999*, volume 1, pages 496–501. IEEE.

[Pollefeys et al., 1998] Pollefeys, M., Koch, R., Vergauwen, M., and Van Gool, L. (1998). Metric 3d surface reconstruction from uncalibrated image sequences. In *3D Structure from Multiple Images of Large-Scale Environments*, pages 139–154. Springer.

[Pons et al., 2007] Pons, J.-P., Keriven, R., and Faugeras, O. (2007). Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193.

[Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition.

[Rosenberg et al., 2006] Rosenberg, I. D., Davidson, P. L., Muller, C. M., and Han, J. Y. (2006). Real-time stereo vision using semi-global matching on programmable graphics hardware. In *ACM SIGGRAPH 2006 Sketches*, page 89. ACM.

[Roy and Cox, 1998] Roy, S. and Cox, I. J. (1998). A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the 6th International Conference on Computer Vision (ICCV) 1998*, pages 492–499. IEEE.

[Seitz et al., 2006] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the Conference on Computer vision and pattern recognition (CVPR) 2006*, pages 519–528. IEEE.

[Seitz and Dyer, 1999] Seitz, S. M. and Dyer, C. R. (1999). Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173.

[Shimizu and Okutomi, 2001] Shimizu, M. and Okutomi, M. (2001). Precise sub-pixel estimation on area-based matching. In *Proceedings of the International Conference on Computer Vision (ICCV) 2001*, pages 90–97 vol.1. IEEE.

[Shimizu and Okutomi, 2002] Shimizu, M. and Okutomi, M. (2002). An analysis of sub-pixel estimation error on area-based image matching. In *Proceedings of the International Conference on Digital Signal Processing (DSP) 2002*, pages 1239–1242.

[Sinha et al., 2012] Sinha, S. N., Kopf, J., Goesele, M., Scharstein, D., and Szeliski, R. (2012). Image-based rendering for scenes with reflections. *ACM Transactions on Graphics*, 31(4):100:1–100:10.

[Sivan and Samet, 1992] Sivan, R. and Samet, H. (1992). Algorithms for constructing quadtree surface maps. In *Proceedings of the International Symposium on Spatial Data Handling*, pages 361–370.

[Slabaugh et al., 2001] Slabaugh, G., Culbertson, B., Malzbender, T., and Schafer, R. (2001). A survey of methods for volumetric scene reconstruction from photographs. In *Proceedings of the Conference on Volume Graphics (VG) 2001*, pages 81–101. Eurographics Association.

[Snavely et al., 2006] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. In *Proceedings of the 33rd annual conference on Computer graphics and interactive techniques (SIGGRAPH) 2006*, SIGGRAPH '06, pages 835–846, New York, NY, USA. ACM.

[Stein et al., 2006] Stein, A., Huertas, A., and Matthies, L. (2006). Attenuating stereo pixel-locking via affine window adaptation. In *Proceedings of the International Conference on Robotics and Automation (ICRA) 2006*, pages 914–921. IEEE.

[Strecha et al., 2008] Strecha, C., von Hansen, W., Van Gool, L., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, pages 1–8. IEEE.

[Sun et al., 2003] Sun, J., Zheng, N.-N., and Shum, H.-Y. (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800.

[Sun et al., 2011] Sun, X., Mei, X., Jiao, S., Zhou, M., and Wang, H. (2011). Stereo matching with reliable disparity propagation. In *Proceedings of the International Conference on3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT) 2011*, pages 132–139. IEEE.

[Tao et al., 2001] Tao, H., Sawhney, H., and Kumar, R. (2001). A global matching framework for stereo computation. In *Proceedings of the International Conference on Computer Vision (ICCV) 2001*, pages 532–539. IEEE.

[Tappen and Freeman, 2003] Tappen, M. and Freeman, W. (2003). Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proceedings of the International Conference on Computer Vision (ICCV) 2003*, pages 900–906. IEEE.

[Tola et al., 2008] Tola, E., Lepetit, V., and Fua, P. (2008). A fast local descriptor for dense matching. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, pages 1–8. IEEE.

[Tomasi and Kanade, 1991] Tomasi, C. and Kanade, T. (1991). *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh.

[Triggs et al., 2000] Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustmenta modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer.

[Turk and Levoy, 1994] Turk, G. and Levoy, M. (1994). Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH) 1994*, pages 311–318. ACM.

[Veksler, 2007] Veksler, O. (2007). Graph cut based optimization for mrfs with truncated convex priors. In *Conference on Computer Vision and Pattern Recognition (CVPR) 2007*, pages 1–8. IEEE.

[Verdie et al., 2015] Verdie, Y., Lafarge, F., and Alliez, P. (2015). Lod generation for urban scenes. *ACM Transactions on Graphics*, 34(3):30:1–30:14.

[Verhoeven, 2011] Verhoeven, G. (2011). Taking computer vision aloft–archaeological three-dimensional reconstructions from aerial photographs with photoscan.

[Viola and Wells III, 1997] Viola, P. and Wells III, W. M. (1997). Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154.

[Vogiatzis et al., 2008] Vogiatzis, G., Torr, P. H., Seitz, S. M., and Cipolla, R. (2008). Reconstructing relief surfaces. *Image and Vision Computing*, 26(3):397–404.

[Vu, 2011] Vu, H. H. (2011). *Large-scale and high-quality multi-view stereo*. PhD thesis, Paris Est.

[Waechter et al., 2014] Waechter, M., Moehrle, N., and Goesele, M. (2014). Let there be color! large-scale texturing of 3d reconstructions. In *Proccedings of the European Conference on Computer Vision (ECCV) 2014*, pages 836–850. Springer.

[Wu, 2007] Wu, C. (2007). Siftgpu: A gpu implementation of scale invariant feature transform (sift).

[Wu, 2011] Wu, C. (2011). Visualsfm: A visual structure from motion system.

[Wuttke et al., 2012] Wuttke, S., Perpeet, D., and Middelmann, W. (2012). Quality preserving fusion of 3d triangle meshes. In *15th International Conference on Information Fusion (FUSION) 2012*, pages 1476–1481. IEEE.

[Xiong and Matthies, 1997] Xiong, Y. and Matthies, L. (1997). Error analysis of a real-time stereo system. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 1997*, pages 1087–1093. IEEE.

[Yang et al., 2003] Yang, R., Pollefeys, M., and Welch, G. (2003). Dealing with textureless regions and specular highlights-a progressive space carving scheme using a novel photo-consistency measure. In *Proceedings of the International Conference on Computer Vision (ICCV) 2003*, volume 1, pages 576–576. IEEE.

[Yu et al., 2007] Yu, T., Xu, N., and Ahuja, N. (2007). Shape and view independent reflectance map from multiple views. *International journal of computer vision*, 73(2):123–138.

[Zabih and Woodfill, 1994] Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV) 1994*, pages 151–158. Springer.

[Zach, 2008] Zach, C. (2008). Fast and high quality fusion of depth maps. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT) 2008*, volume 1. Citeseer.

[Zach et al., 2007] Zach, C., Pock, T., and Bischof, H. (2007). A globally optimal algorithm for robust tv-l 1 range image integration. In *Proceedings of the International Conference onComputer Vision (ICCV) 2007*, pages 1–8. IEEE.

Acknowledgements

Herewith I gratefully thank all people who contributed to the compilation of this thesis. First, I would like to thank Prof. Fritsch for giving me the opportunity to work and conduct my thesis with the IfP. Moreover, I would like to thank Prof. Van Gool for investing his time and agreeing to be the co-examiner of this thesis. Many thanks to Prof. Haala for all the valuable input and help within my years working with the IfP. Furthermore, big thanks go to my colleagues at the IfP and nFrames for all their help, fruitful discussions and providing a nice working atmosphere. Last but not least I would like to thank my family and friends for their unconditional support and encouragement within the last years.

Ciriculum Vitae

Personal

Name	Mathias Rothermel
Date of birth	05.09.1982
Place of birth	Stuttgart, Germany

Education

2003 - 2009	Diplom, Technical Cybernetics, Stuttgart University
1993 - 2002	Abitur, Gymnasium Renningen

Experience

03.2008 - 06.2008	Intern, Robert Bosch GmbH, Leonberg
07.2008 - 09.2009	Research Intern, Robert Bosch RTC, Palo Alto
02.2010 - 12.2014	Research Assistant, IfP Stuttgart University, Stuttgart
01.2015 - current	Co-Founder and CTO, nFrames GmbH, Stuttgart