Technische Universität München

Fakultät für Informatik
Lehrstuhl für Bildverarbeitung und Künstliche Intelligenz

Efficient Large-Scale Stereo Reconstruction using Variational Methods

Georg Kuschk

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Matthias Nießner
Prüfer der Dissertation:	1. Prof. Dr. Daniel Cremers
	2. Prof. Dr. Michael Möller Universität Siegen

Die Dissertation wurde am 31.10.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 13.03.2019 angenommen.

Abstract

This thesis investigates the use of convex variational methods for depth reconstruction from optical imagery and fusion of multiple depth maps into combined depth maps with higher accuracy. Dense depth reconstruction from two or more camera views are an important subject of research in computer vision, since measurement density is much higher than other depth sensing techniques, namely active depth sensing via infrared pattern projection or Lidar and Radar based techniques - even though the latter ones are more accurate and robust in depth. Other advantages of cameras are their low costs and low power consumption due to their passive sensing principle. Approaches are ranging from autonomous driving cars, obstacle avoidance or surveying UAVs up to detailed reconstruction of remote terrains using space borne imagery.

In particular, we propose a fast algorithm for high-accuracy large-scale outdoor dense stereo reconstruction. To this end, we present a structure-adaptive second-order Total Generalized Variation (TGV) regularization which facilitates the emergence of planar structures by enhancing the discontinuities along building facades. Instead of solving the arising optimization problem by a coarse-to-fine approach, we propose a quadratic relaxation approach which is solved by an augmented Lagrangian method. This technique allows for capturing large displacements and fine structures simultaneously.

For the application in autonomous driving, we further present an algorithm for dense and direct large-scale visual SLAM that runs in real-time on a commodity notebook. We developed a fast variational dense 3D reconstruction algorithm which robustly integrates data terms from multiple images thus enhancing quality of the image matching. An additional property of this variational reconstruction framework is the ability to integrate sparse depth priors (e.g. from RGB-D sensors or LiDAR data) into the early stages of the visual depth reconstruction, leading to an implicit sensor fusion scheme for a variable number of heterogenous depth sensors. Embedded into a keyframe-based SLAM framework, this results in a memory efficient representation of the scene and therefore (in combination with loop-closure detection and pose tracking via direct image alignment) enables us to densely reconstruct large scenes in real-time.

Finally, applied to space-borne remote sensing, we present an algorithm for robustly fusing digital surface models (DSM) with different ground sampling distances and confidences, using explicit surface priors to obtain locally smooth surface models. The optimization using L_1 based differences between the separate DSMs and in-

corporating local smoothness constraints is also inherently able to include weights for the input data, therefore allowing to easily integrate invalid areas, fuse multi-resolution DSMs and to weigh the input data.

Zusammenfassung

Diese Arbeit untersucht die Verwendung von konvexen Variationsmethoden für die Tiefenrekonstruktion anhand optischer Bilder und die Fusion mehrerer Tiefenkarten in kombinierte Tiefenkarten mit höherer Genauigkeit. Eine dichte Tiefenrekonstruktion aus zwei oder mehr Kameraansichten ist ein wichtiges Thema der Computerbildverarbeitung, da die Messdichte viel höher ist als bei anderen Tiefenmesstechniken, insbesondere aktiven Tiefenmessungen mittels Infrarotmusterprojektion oder Lidar- und Radar-basierte Techniken - auch wenn diese genauer und robuster in der Tiefenmessung sind. Weitere Vorteile von Kameras sind ihre geringen Kosten und ihr geringer Stromverbrauch aufgrund ihres passiven Messprinzips. Die Ansätze reichen von autonom fahrenden Autos über die Vermeidung von Hindernissen oder die Vermessung durch UAVs bis hin zur detaillierten Rekonstruktionen von abgelegenen Gebieten mit Hilfe von weltraumgestützten Bildaufnahmen.
Insbesondere schlagen wir einen schnellen Algorithmus für eine hochgenaue, dichte Stereo-Rekonstruktion für großräume Outdoor-Szenarien vor. Zu diesem Zweck präsentieren wir eine strukturadaptive TGV-Regularisierung (Total Generalized Variation) zweiter Ordnung, welche die Entstehung planarer Strukturen durch die Verbesserung der Diskontinuitäten entlang von Gebäudefassaden erleichtert. Anstatt das entstehende Optimierungsproblem durch einen Coarse-to-Fine-Ansatz zu lösen, schlagen wir einen quadratischen Relaxationsansatz vor, der durch eine Augmented Lagrange Methode gelöst wird. Mit dieser Technik können große Verschiebungen naher Objekte im Bildbereich wie auch feine Strukturen gleichzeitig erfasst werden.
Für die Anwendung im autonomen Fahren stellen wir außerdem einen Algorithmus für dense und direct large-scale visual SLAM vor, das in Echtzeit auf einem Standard-Notebook läuft. Wir haben einen effizienten, variationsbasierten dichten 3D-Rekonstruktionsalgorithmus entwickelt, der Daten aus mehreren Bildern robust integriert und somit die Qualität des Bildmatchings verbessert. Eine zusätzliche Eigenschaft dieses variationsbasierten Rekonstruktionsframeworks ist die Fähigkeit, dünnbesetzte Tiefen a-priori Informationen (z.B. von RGB-D-Sensoren oder LiDAR-Daten) in die frühen Stadien der Rekonstruktion der visuellen Tiefe zu integrieren, was zu einem impliziten Sensorfusionsschema für eine variable Anzahl heterogener Tiefensensoren führt. Eingebettet in ein Keyframe-basiertes SLAM-Framework führt dies zu einer speichereffizienten Darstellung der Szene und ermöglicht somit (in Kombination mit Loop-Closure-Erkennung und Pose-Tracking über direct image alignment) die dichte Rekonstruktion von umfangreichen Szenen in Echtzeit.

Schließlich stellen wir einen Algorithmus zur robusten Fusionierung von digitalen Oberflächenmodellen (DSM) mit verschiedenen Bodenabtastungsabständen und Konfidenzen vor, wobei explizite Oberflächen priors verwendet werden, um lokal glatte Oberflächenmodelle zu erhalten. Die Optimierung unter Verwendung von auf der L_1-Norm basierenden Differenzen zwischen den einzelnen DSMs und dem Einbeziehen lokaler Glattheitseinschränkungen ist auch inhärent in der Lage, Gewichtungen für die Eingabedaten einzuschließen, wodurch es auch möglich ist, ungültige Bereiche ohne vorliegende Messpunkte in den Optimierungsprozess zu integrieren, DSMs mit mehreren verschiedenen Bodenabtastungsabständen zu fusionieren und die Eingabedaten allgemein zu gewichten.

Acknowledgements

First of all, I would like to thank my doctoral advisor Prof. Daniel Cremers for giving me the opportunity to pursue my academic research under his supervision. My second referee Michael Möller deserves just as much appreciation. Both Daniel and Michael inspired and helped me a lot with excellent and fruitful discussions.
During my time at the German Aerospace Center (DLR) I worked with many great people who also deserve huge credit. Namely I want to thank my colleagues and co-authors Pablo d'Angelo, Peter Reinartz, Thomas Krauss as well as my colleagues Peter Fischer, Oliver Meynberg, Ke Zhu, Jiaojiao Tian and my students Grigory Shelekov, Marina Schicht, Ksenia Davydova, David Gaudrie.
During my short time at the university of Graz (TUG) I was honored to make the acquaintance with Thomas Pock, whose inspirational sessions at the whiteboard were marvelous. Many thanks also go to Gottfried Graber and René Ranftl from whom I learned a lot of practical implementation details and tricks.
I also would like to thank all of my colleagues, co-authors, contributors and friends at the Technical University of Munich (TUM): Mohamed Souiai, Frank Steinbrücker, Thomas Möllenhoff, Jan Stühmer, Robert Maier, Thomas Windheuser and my students during that time, all of them pursuing an interesting career now: Aljaz Bozic, Björn Häfner and Dennis Mack.
Finally and foremost, many thanks got to my family for raising and supporting me as well as encouraging me all the time to follow my passion regardless the obstacles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With advancing automation, large-scale 3D reconstruction is increasingly becoming more important in various scientific fields as well as in common life. Applications are ranging from autonomous driving of cars or mobile robots in general, flight planning for unmanned aerial vehicles or remote sensing based digital surface reconstruction for wide-area physical simulations like flood simulation, propagation of radio beams, 3D change detection etc. Despite the advances of modern *active* depth-sensing technologies like Lidar, Radar, Time-of-Flight and projector-camera systems, depth estimation based on cameras only still is on par with these active sensing technologies.
The advantages of a very high spatial resolution, dense measurements, the absence of interferencing problems due to passive sensing, cheap costs and low power consumption render cameras as a very attractive sensor for this problem. From a philosophical point of view, one could even argue, that most higher biological life uses optical systems for navigation, thereby implying that evolution proved this a good choice for environmental sensing. Camera-only systems of course have their own shortcomings for 3D reconstruction (aperture problem / repetitive textures, textureless image regions, specular reflections, semi transparent surfaces etc). This directly implies a smart combination and fusion of different sensor modalities, thereby mitigating their respective shortcomings, in cases where this is applicable.
So despite being an active research area for decades, the need for more accurate and robust results as well as computationally cheap approaches still drives investigation in the field of optical 3D reconstruction. In this work the aspect of obtaining very fast dense depth reconstruction, while simultaneously achieving high accuracy, is being investigated. Apart from the reconstruction process itself, fusing image- and depth information from multiple images is examined as well. This work attempts to meet the demands of two somewhat different applications:

- Far-range satellite based depth reconstruction.
- Close-range depth reconstruction in automotive scenarios and

While the automotive application has very strict real-time requirements and operates on images the size of $10^6 - 10^7$ pixel, captured with 10-30 frames per second, the remote sensing application typically consists of only 1-3 captured images of the scene, with a size of roughly 10^9 pixel per image.
For the automotive scenario, this results in very short inter-frame distances, both temporarily and spatially, making this use case applicable for fusing different depth maps to a single depth map exhibiting higher accuracy. Due to the short baseline between captured images, perspective changes only slightly between images, thereby simplifying the problem of matching image areas between two frames.
For remote sensing imagery arising from satellites or aerial sensors, the baseline between two captured images is typically very large (15 - 60°), resulting in quite different reflection properties of the captured surfaces.

Motivation

As a practical motivation Figure 1.1 shows an application of large scale 3D reconstruction based on close-range sensing mobile cameras, e.g. in the case of autonomous driving. In these scenarios, where thousands of successive depth maps are generated with the mobile platform dynamically moving around the scene, important aspects like drifting position estimation (compared to ground truth) and revisiting of identical places have to be considered by applying an overall SLAM framework. We will discuss this in detail in Chapter 4.
Different sensor technologies exist, nowadays namely Lidar sensors, which have a higher longitudinal accuracy, but suffer from sparsity and missing texture information. Also in Chapter 4 we propose a strategy for depth estimation from camera with the optional help of integration existing sparse depth data early in the reconstruction process.

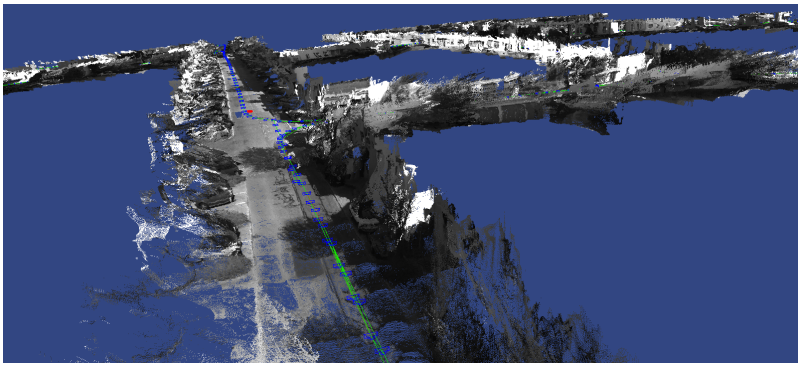
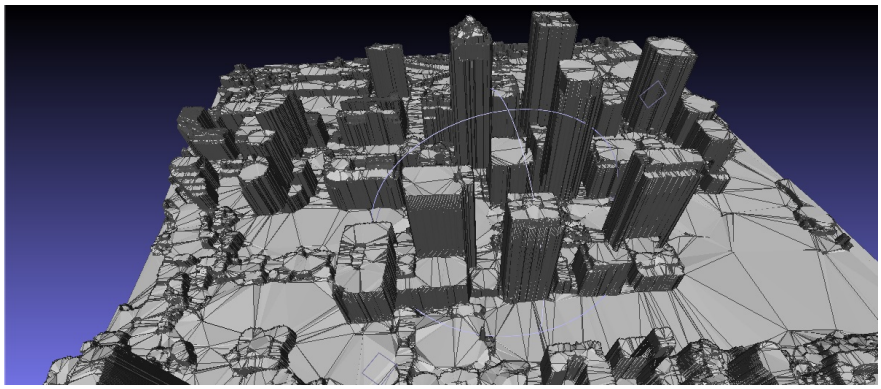
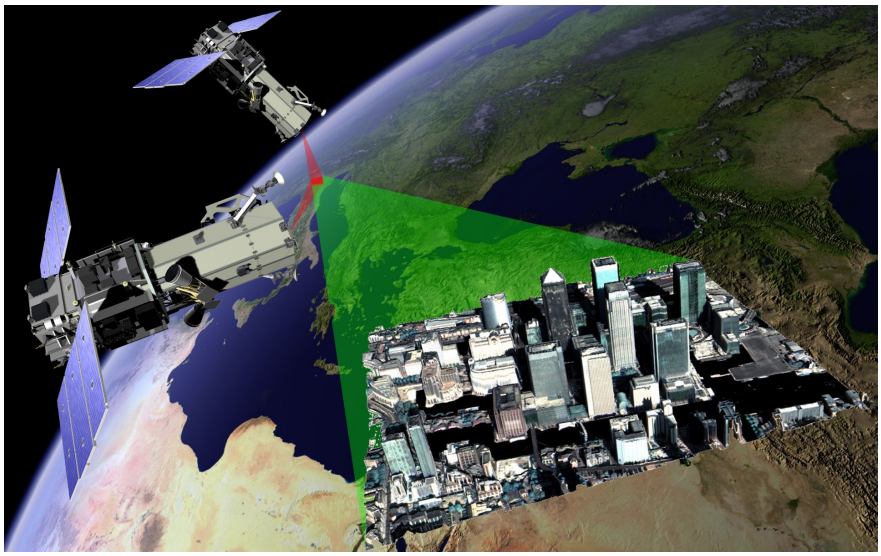


Figure 1.1: Resulting large scale 3D reconstruction of a complete SLAM framework in automotive driving scenarios - as will be described in Chapter 4.

Figure 1.2 gives an overview of remote sensing based surface reconstruction and some exemplary applications. While radar satellites - based on *Synthetic Aperture Radar* (SAR) like TanDEM-X and TerraSAR-X do not suffer from clouds occluding the observed surface and exhibiting high longitudinal accuracy, their lateral resolution and accuracy is very low compared to optical satellites and of course surface texture cannot be captured as well. An obvious approach would be fusing the corresponding 3D information while retaining the advantageous properties of each input data. In Chapter 5 we will have a detailed look at how to accomplish this fusion.



(a) 3D reconstruction for autonomous flight planning

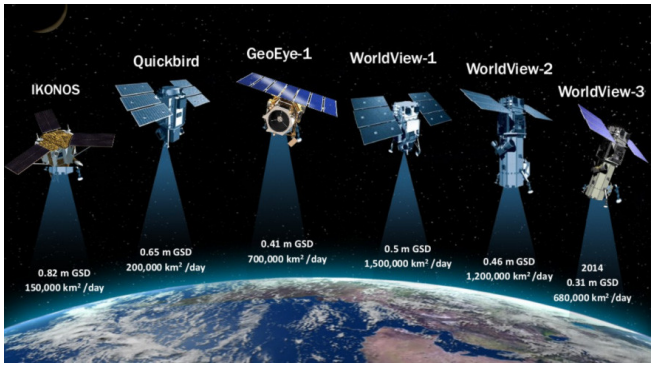


(b) Space-borne 3D reconstruction

Figure 1.2: Satellite based 3D reconstruction, serving exemplary applications like change detection in restricted areas or unchartered, sprawling mega cities, like flood simulations, propagation of radio beams, autonomous flight planning etc.

Remote sensing data

As spaceborne remote sensing data is not readily available for most computer vision researchers, we would like to give a short introduction over the nature of corresponding satellite data. A detailed explanation of the image capturing process and the camera model is given in Section 2.1.2.



(a) Worldview satellites, image credit `https://www.digitalglobe.com`

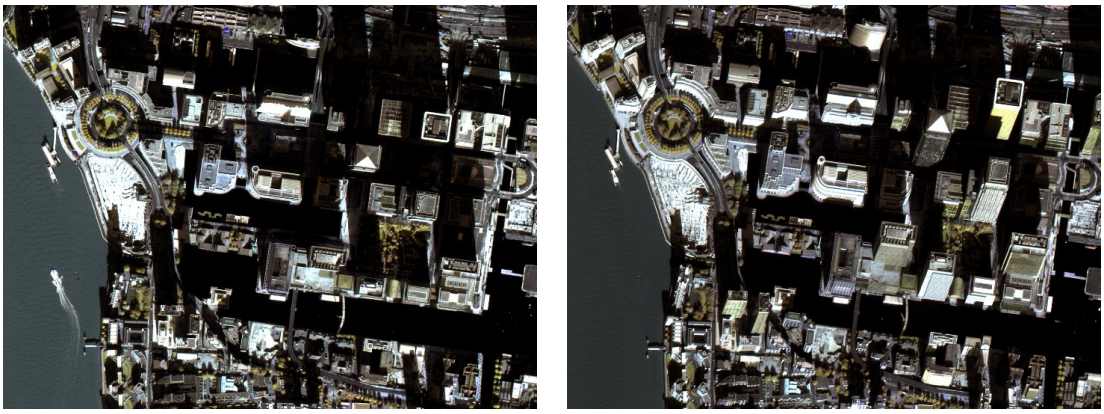
(b) One of the Pleiades satellites, image credit `https://pleiades.cnes.fr`

Figure 1.3: Overview of some common commercial optical satellites. Each satellite produces data of approximately 1 terabyte per day with images having a size of around 1 gigapixel and a ground sampling distance (GSD) of 0.3-0.8m.

Even though the main use case is 2D monitoring of land surfaces, the already captured image data can be readily used for 3D reconstruction processes. However, due to the altitude of roughly 700km to the observed surface, it is practically not possible to capture images with a satellite mounted stereo camera from one specific position in orbit and perform 3D reconstruction on these. Instead - to get a convenient baseline for the underlying triangulation process - images from different position (and therefore different time steps) are used. Since the 3D reconstruction process typically involves a static world assumption, this leads to problems with large moving objects as can be seen in Figure 1.4.
Geolocation accuracy (= pose estimation) of the satellites is typically in the range of 3m, if ground control points (known reference points in the image) are given accuracy is in range of 1m. Pose estimation of satellites is done using a start tracker as an optical device to identify and measure the position of given stars and inertial measurement units (IMU) consisting of gyroscopes and accelerometers. Life expectancy (based on decommissioned satellite like IKONOS and Quickbird) is typically in the range of 7-13 years.

	launch	altitude	bit depth	panchromatic GSD	multispectral channels and GSD
Cartosat-1	2005	620 km	10	2.5 m	-
GeoEye-1	2008	681 km	11	0.5 m	4 (2.0m)
Worldview-2	2009	770 km	11	0.5 m	8 (2.0m)
Pleiades	2012	695 km	12	0.5 m	5 (2.0m)
Worldview-3	2014	617 km	11	0.3 m	8 (1.2m)

Table 1.1: Basic data of most common operational commercial optical satellites



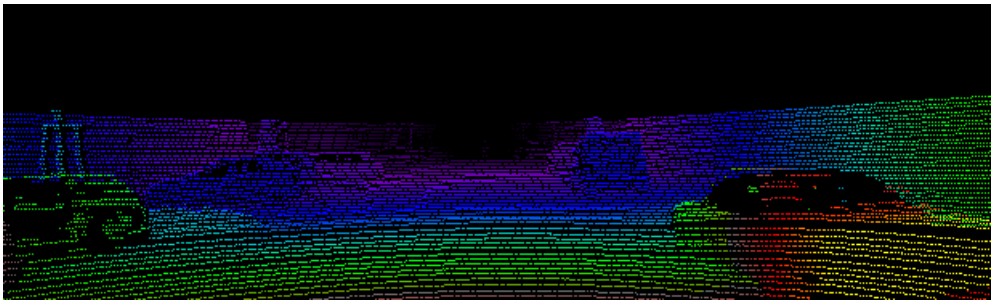
(a) Satellite position at time step t (b) Satellite position at time step $t+1$

Figure 1.4: Exemplary Worldview-2 satellite imagery. Please note the strong perspective change between consecutive images (slanted buildings) and temporal (moving boats) change, as well as large occlusion areas between the skyscrapers. Despite a bit depth of 11 Bit the shadows have a very low signal-to-noise ratio.

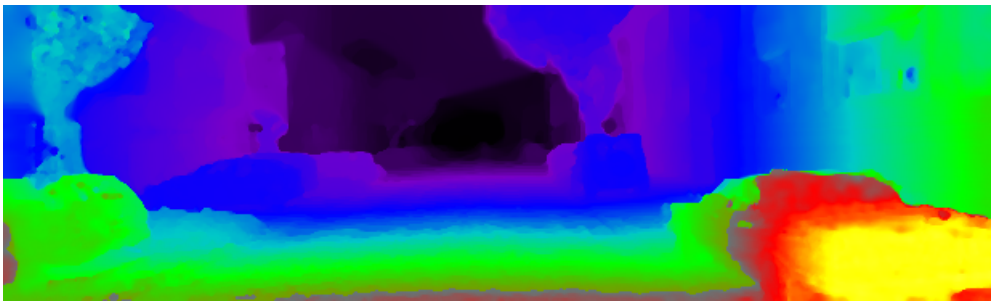
Automotive data

For the application of depth sensing in highly-automated driving scenarios, we use the well-known and publicly available KITTI data [34], which for our tasks consists of two automotive stereo camera systems (RGB and gray) and a 64-layer laser scanner. See Figure 1.5 for exemplary data.

(a) Left camera frame



(b) 3D Lidar data projected into 2D image space



(c) Dense depth reconstruction

Figure 1.5: Exemplary imagery and Lidar data for urban driving scenarios as in [34], plus dense depth reconstruction as described in Chapter 4

Literature Overview

Datasets

Despite earlier research, the publication of the Middlebury Stereo Benchmark [99], [100] marked a milestone in the development of depth reconstruction algorithms, allowing evaluation and comparison of the evolving ideas and algorithms on a standardized benchmark instead of judging the quality on non-public and withheld data. Over the next years, with continuously improving reconstruction quality the dataset was extended multiple times to incorporate more challenging data [48] and [98] (see Figure 1.6). Still the acquired data covered only indoor scenarios due to the ground truth depth acquisition system based on structured light and continued being restricted to 2-view stereo imagery.
As a logical extension, and with the advent of more and more automotive scenarios in industry, real-world outdoor datasets like the famous KITTI dataset were published [34], [71], [54], covering challenging outdoor scenarios as well. As of now, the corresponding ground truth is generated using a calibrated and synchronized laser scanning system.
With the upcoming success of deep learning approaches, the need for training data grew exponentially, thus training data generation became very expensive. This lead to the generation of various synthetic datasets, such as [16], [68], [118], [95], [94].

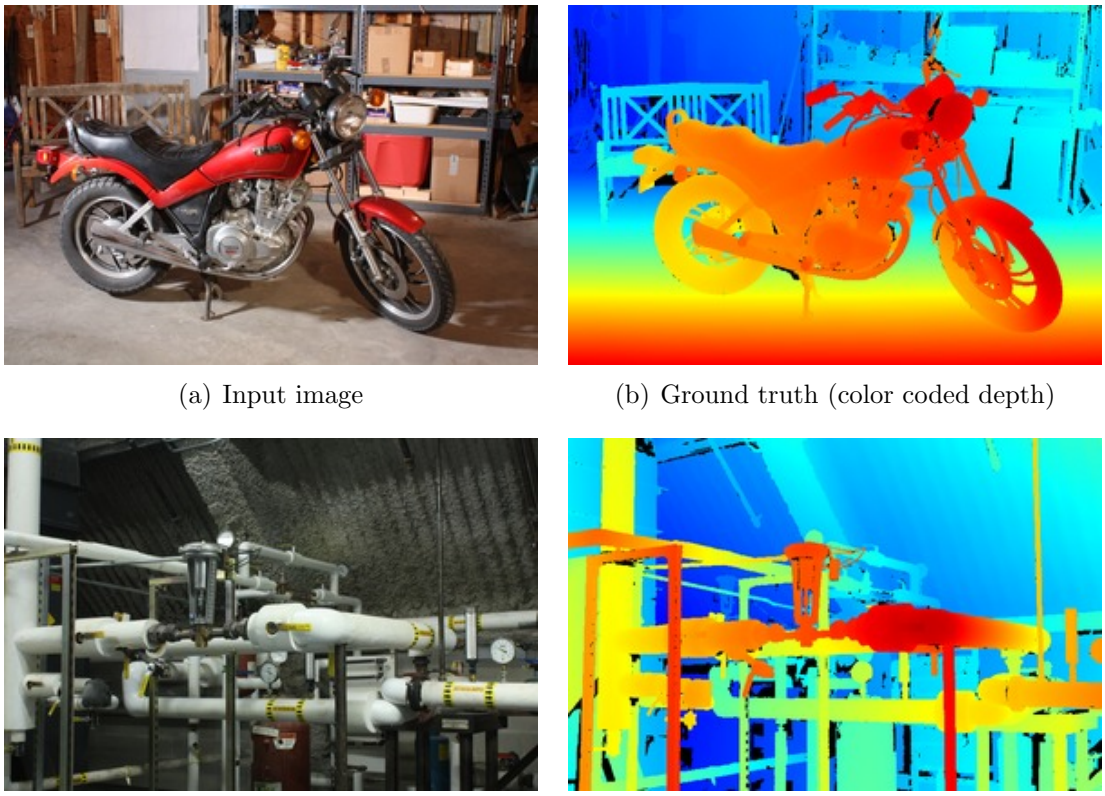
Stereo depth estimation

An overview of the most commonly used datasets for evaluation of depth reconstruction is given in Table 1.2. The number of submissions indicates a vivid development and active research in this field, making a complete overview of depth reconstruction algorithms intractable in this chapter. We rather give a short overview of a group of algorithms which dominated the field of depth reconstruction by their respective time.

Dataset	Number of submissions in benchmark
Middlebury 2003 [100]	167
Middlebury 2014 [98]	71
KITTI 2012 [34]	100
KITTI 2015 [71]	84

Table 1.2: Most commonly used datasets for evaluation of depth reconstruction

Finding the best-matching image patch in other images for a given pixel position in a reference image, together with the corresponding camera positions, allows for



(a) Input image (b) Ground truth (color coded depth)

(c) Input image (d) Ground truth (color coded depth)

Figure 1.6: Exemplary data from the Middlebury dataset [98]

triangulation and therefore estimating the depth of that particular pixel. In Section 2.1 and 2.1.2 we will go further into detail, but we will highlight the state of the art of image matching cost functions in this section.
Due to perspective changes, illumination changes, occlusions etc, image matching itself is not trivially leading to the optimal solution and much research was done investigating proper image matching cost functions. Very simple cost functions with no spatial support are for example AD (absolute difference), Birchfield-Tomasi measure [8] or Mutual Information [116].
Comparing the intensity values of two corresponding pixels alone is prone to noise and ambiguities (see Figure 2.13) and therefore more local image information needs to be incorporated into the image matching cost functions. The two opposing principles which must be met are a large spatial support for generating descriptive and discriminative features vectors on the one hand and a small spatial support on the other hand, as spatial support implies assumptions of the depth of neighboring pixels, leading to local planarity assumption.

To restrict computational complexity, fronto-parallel assumption of the neighboring pixels (all 3D points lying on a plane parallel to the image plane) is the standard case resulting in cost functions comparing image patches around the pixel position using sum of absolute difference (SAD), sum of squared differences (SSD), normalized cross correlation (NCC), Census transform [125] etc. For high accuracy cases, evaluating the cost function on multiple (slanted) planes is possible as well [33], [10]. The aforementioned Census transform is a very powerful image descriptor for dense image matching because it is robust to many forms of illumination changes (to some degree even non-linear ones such as specularities) and at the same time computationally very cheap. Naturally. different work was done trying to improve the descriptive and discriminative strength while at the same time maintaining the speed. Replacing the binary comparison with a ternary decision was done in [105], increasing robustness against noise, especially at the central pixel. In [32] intensities are not compared with the central pixel intensity directly but with the mean value of its 3×3 neighborhood, also increasing robustness against noise. A scale-robust Census matching was introduced in [88], based on a circular sampling strategy with different radii corresponding to different scales and taking the minimum cost over these different scales.
A common approach to further strengthen the image matching cost function is local cost aggregation based on pixel similarities in terms of color and distance as described in the seminal work of Adaptive support-weights [124] which will be also described in detail in Section 2.2.
Of the advanced and more descriptive features (mainly used in image recognition) like SIFT [67], SURF [2], BRIEF [17], BRISK [64] only the DAISY descriptor [114], designed for speed, showed a somewhat acceptable speed for using it in dense stereo reconstruction. Due to some invariance against rotation, scaling and simple radiometric changes, using the DAISY descriptors (or any other of the abovementiond features) yields very good results for wide-baseline stereo like in satellite based 3D reconstruction, where some simple dissimilarity measures fail because of the large perspective differences.
For more detailed evaluations of different cost functions, we refer to the surveys of e.g. [99], [112], [48], [49].

Regularization

Despite much work on the image matching cost functions, estimation of the depth of a pixel is still largely independent of its neighboring pixels, leading to noisy cost functions and wrong depth estimation due to local minima (see Figure 2.13 and 2.3). To reduce noise in the resulting depth map a very simple approach is to run a 2D median filtering on the depth map, replacing outliers by the median depth in a local support window. This approach however only partly cures the symptoms on much reduced data - instead of addressing the original problem earlier with all input data

(information about the image matching cost functions especially) still available. To this end, instead of estimating the depth of each pixel solely on its image matching information, additional priors on the resulting 3D surfaces are enforced in the overall reconstruction process as described in detail in Section 2.1. These priors are typically called smoothness terms, referring to the effect of favoring locally planar surfaces in the 3D reconstruction. To do so, they enforce pixels in a local neighborhood to have similar values for depth. However, computing the optimal solution of such a reconstruction problem becomes in general extremely hard because of the implied computational demand and combinatorial complexity. Thus, many approximation techniques have been investigated, with the goal of having a guaranteed local optimum close to the global one while being as fast as possible. A detailed overview of the results of the past two decades of research spent in this field is out of scope of this thesis. So we limit ourselves to a short overview of a group of algorithms which dominated the field of depth reconstruction by their respective time.

Markov Random Fields [13], [50], [12], [53] and Belief Propagation [111], [123], [122] were the first family of such algorithms, dominating the state-of-the-art at their time and in simple special cases even guaranteeing to result in a global optimum. The basic idea of Markov Random Fields is posing the depth reconstruction (and other image processing tasks) as labeling problem, assigning each pixel a label corresponding to its depth, while at the same time taking into account the interaction between neighboring pixels, namely enforcing similar labels in a local neighborhood. While this labeling from a combinatorial point of view is NP-hard, the seminal paper of [13] framed the problem as a specialized graph for the energy function which then can be approximately minimized via repeatedly applying efficient max-flow min-cut algorithms. An overview of such max-flow min-cut algorithms can be found in [9].

Semi Global Matching (SGM) was developed shortly afterwards in the work of [44], [45], [48], [46],[49], [47]. Instead of tackling the hard 2-dimensional problem, this remarkably efficient family of algorithms is iteratively passing over the image in 1-dimensional sweeps from different directions and solving the separate 1D problems via efficient and accurate dynamic programming. The result after each such directional scan is passed as approximate solution to the next directional pass where it is again refined. SGM might not be the most accurate algorithm, but because of it's low computational complexity it is widely used for real-time depth reconstruction on low-cost hardware.

Variational Methods were brought to interest by the seminal paper of Rudin, Osher and Fatemi [96], who introduced total variation for image denoising. Similar to Markov Random Fields, minimizing the total variation of a functional defined over image space is done iteratively, with a spatially limited direct interaction of variables. Instead of enforcing smooth discrete depth-labels to neighboring pixels, variational methods inherently minimize on a continuous label space by solving the differential equations arising from minimizing the first order derivative of the depth

map (minimizing jumps in depth between neighboring pixels). With the advent of modern general purpose graphics processing units (GPGPU or just GPU), these highly parallelizable algorithms became computationally feasible, thereby boosting research in this field [19],[21], [84], [83], [85] culminating in the celebrated paper introducing the primal-dual algorithm [22].
Providing a general framework and an efficient solution including convergence guarantees for a large number of optimization problems, this work is also the underlying foundation of this thesis. For a very detailed overview of applying Total Variation for image analysis we refer to the well written report of [20]. In a follow-up work [82] the primal-dual algorithm was further accelerated by using local step-sizes for the gradient descent based optimization steps instead of the former global step size. Although the primal-dual algorithm is applicable to a large number of optimization problems, it was already used for depth reconstruction in the original paper [22], with a large number of follow-up work like [36], [89], [90].
The original work contained a regularization term for the depth reconstruction by directly minimizing the sum of gradients of the resulting depth map, thereby favoring fronto-parallel surfaces. In [14] and [87] this constraint was lifted and extended to higher order regularizers by the so-called Total Generalized Variation (TGV). Especially for depth reconstruction the 2nd order TGV has a major impact by favoring slanted locally planar surfaces in the scene.
An anisotropic version of the regularizers based on the Nagel-Enkelmann operator [74] was further introduced and used for TV [120] as well as for TGV [89]. Tackling the problem of over-smoothing image areas with large depth changes, the smoothing amount could now be steered according to additional image cues (like e.g. edge information).
In [119] and [88] the Markov property of image pixels only interacting directly with their direct adjacent neighbors was relaxed by allowing direct interactions over further distances (Non-Local Total Variation). As a downside of this, memory consumption increases and convergence speed drops significantly.
Aside from the depth reconstruction application, variational approaches can also readily be applied to denoising and fusing multiple low-quality depth maps which will be detailed in Section 5, building up or having similarity to the work of [127], [126], [87], [30], [80].

Contributions of this Thesis

This thesis focuses and summarizes the work presented in [59], [58], [**?**], which is the result of the joint work with Pablo d'Angelo, David Gaudrie, Aljaž Božič, Prof. Peter Reinartz and Prof. Daniel Cremers. Closely related work was additionally presented in [56], [57], [60] and a complete list of all publications published throughout the period of this thesis is given in Table 1.3. All included papers

are peer-reviewed publications and were published in international conferences or journals.

In chapter 3, we propose a fast algorithm for high-accuracy large-scale outdoor dense stereo reconstruction. To this end, we present a structure-adaptive second-order Total Generalized Variation (TGV) regularization which facilitates the emergence of planar structures by enhancing the discontinuities along building facades. Instead of solving the arising optimization problem by a coarse-to-fine approach, we propose a quadratic relaxation approach which is solved by an augmented Lagrangian method. This technique allows for capturing large displacements and fine structures simultaneously.

For the application in autonomous driving, we further present an algorithm for dense and direct large-scale visual SLAM in Chapter 4 that runs in real-time on a commodity notebook. We developed a fast variational dense 3D reconstruction algorithm which robustly integrates data terms from multiple images thus enhancing quality of the image matching. An additional property of this variational reconstruction framework is the ability to integrate sparse depth priors (e.g. from RGB-D sensors or LiDAR data) into the early stages of the visual depth reconstruction, leading to an implicit sensor fusion scheme for a variable number of heterogenous depth sensors. Embedded into a keyframe-based SLAM framework, this results in a memory efficient representation of the scene and therefore (in combination with loop-closure detection and pose tracking via direct image alignment) enables us to densely reconstruct large scenes in real-time.

In Chapter 5, applied to space-borne remote sensing, we present an algorithm for robustly fusing digital surface models (DSM) with different ground sampling distances and confidences, using explicit surface priors to obtain locally smooth surface models. The optimization using L_1 based differences between the separate DSMs and incorporating local smoothness constraints is also inherently able to include weights for the input data, therefore allowing to easily integrate invalid areas, fuse multi-resolution DSMs and to weight the input data.

Own Publications

	Authors	Title	Publication medium
[58]	Kuschk *et al.*	Real-time Variational Stereo Reconstruction with Applications to Large-scale Dense SLAM	IEEE Intelligent Vehicles Symposium, 2017
[61]	Kuschk *et al.*	Spatially Regularized Fusion of Multiresolution Digital Surface Models	IEEE Transactions on Geoscience and Remote Sensing, 2017
[26]	Davydova *et al.*	Consistent Multi-View Texturing of Detailed 3D Surface Models	ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences, 2015
[55]	Krauss *et al.*	3D-Information Fusion from Very High Resolution Satellite Sensors	Proceedings of International Symposium on Remote Sensing of Environment (ISRSE), 2015
[62]	Kuschk *et al.*	DSM Accuracy Evaluation for the ISPRS Commission I Image Matching Benchmark	ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014
[25]	d'Angelo *et al.*	Evaluation of Skybox Video and Still Image Products	ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014
[93]	Reinartz *et al.*	Advances in DSM Generation and Higher Level Information Extraction from High Resolution Optical Stereo Satellite Data	European Association of Remote Sensing Laboratories (EARSeL), 2014
[59]	Kuschk *et al.*	Fast and Accurate Large-scale Stereo Reconstruction using Variational Methods	ICCV Workshop on Big Data in 3D Computer Vision, 2013
[57]	Kuschk *et al.*	Model-Free Dense Stereo Reconstruction for Creating Realistic 3D City Models	Joint Urban Remote Sensing Event (JURSE), 2013
[56]	Kuschk *et al.*	Large Scale Urban Reconstruction from Remote Sensing Imagery	International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), 2013
[72]	Meynberg *et al.*	Airborne Crowd Density Estimation	ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2013
[60]	Kuschk *et al.*	Fusion of Multi-Resolution Digital Surface Models	ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2013
[24]	d'Angelo *et al.*	Dense Multi-View Stereo from Satellite Imagery	IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2012

Table 1.3: Peer-reviewed publications associated to this thesis

Thesis Outline

This cumulative thesis is structured into six chapters.
In Chapter 1 we give an introduction and motivation of this thesis, providing an overview of relevant literature, as well as giving an overview of the research papers that were published during this thesis.
Chapter 2 presents an overview of the mathematical background in general, describes the involved camera models, stereo image matching functions and regularization techniques as well as a general framework for numerical optimization of the developed models.
In Chapter 3, 4 and 5 we present our work of [59], [58], [**?**] respectively in detail.
We conclude this thesis with Chapter 6, summarizing our research and giving an outlook towards future research possibilities.

Chapter 2

Total Variation based Dense Stereo

Mathematical Preliminaries

Notation

Let the image space of an image I be denoted as $\Omega \subset \mathbb{R}^2$. The image I is a function, mapping a 2-dimensional location to a grayscale or color value

$$I : \Omega \to \mathbb{R}^c \tag{2.1}$$

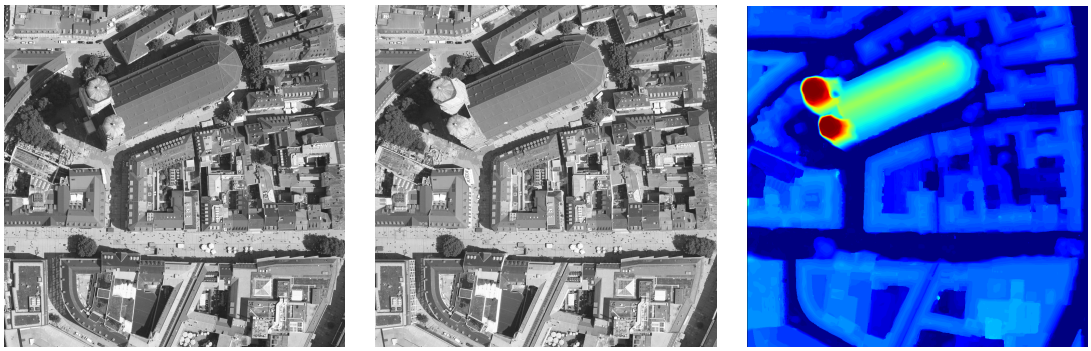
where $c = 3$ usually corresponds to RGB color space and $c = 1$ corresponds to grayscale values. When talking about discrete pixel positions $\mathbf{x}$, the color or intensity values of the image a this position is denoted as $I(\mathbf{x})$.
Our main goal in this thesis is to compute a depth map $\mathbf{u}$, mapping every pixel of a reference input image to a scalar depth value as depicted in Figure 2.1

$$\mathbf{u} : \Omega \to \mathbb{R}. \tag{2.2}$$

In classical dense stereo reconstruction, for every pixel $\mathbf{x} = (x, y)^T \in \Omega$ of the reference image I_{ref} and a number of depth hypotheses $d_i \in [d_{min}, d_{max}]$ with $i \in \{1, .., D\}$, a matching cost is computed by back-projecting the pixel $\mathbf{x}$ into 3D space given the corresponding depth hypothesis d_i, projecting the resulting 3D point into the second image I_2 and obtaining the pixel coordinate $\mathbf{x}' = (x', y')^T$. Finally image information of the two images at their corresponding positions is compared and a matching score computed. The result is the so called *cost volume* [11], containing the *raw* matching costs (Figure 2.2).

The optimal 3D reconstruction given this cost volume is obtained by fitting a 3D surface through this cost volume, having minimal cost (or energy E) in total

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} E(\mathbf{u}) \tag{2.3}$$



(a) Second image I_2 (b) Reference image I_1 (c) Color coded depth map $\mathbf{u}^*$ corresponding to I_1

Figure 2.1: 2-view stereo problem with computed depth map

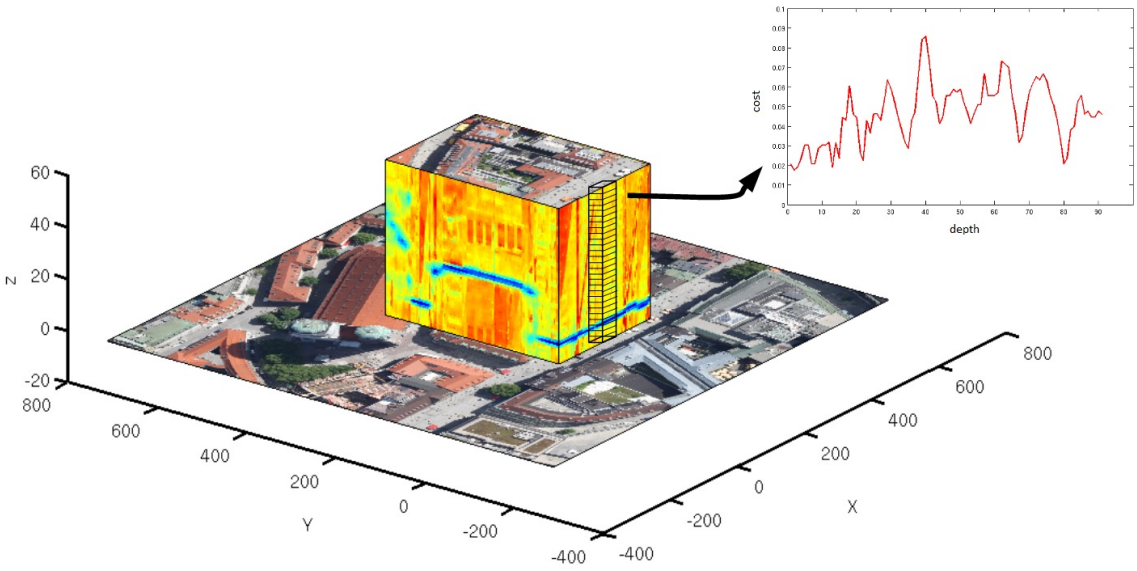
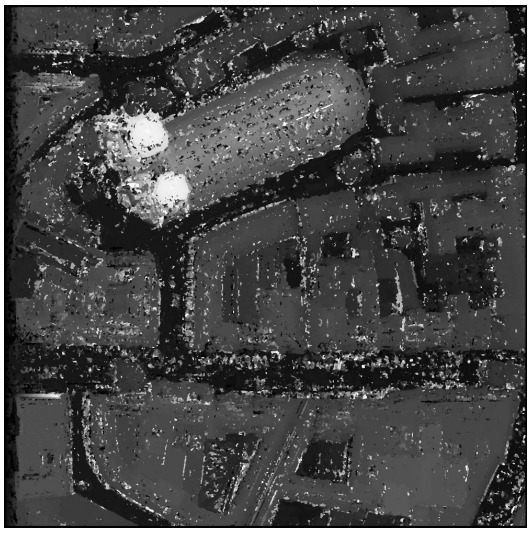


Figure 2.2: Illustration of a Cost volume. For each pixel a number of depth hypotheses are evaluated using image matching, resulting in a 1D cost function for each pixel.

If this energy is expressed solely by its image matching costs, the optimization problem to solve is called *Winner-takes-all* and is given by

$$\begin{aligned} \mathbf{u}^* &= \arg\min_{\mathbf{u}} E_{data}(\mathbf{u}) \\ &= \arg\min_{\mathbf{u}} \int_{\Omega} C(\mathbf{x}, \mathbf{u}) \, \mathrm{d}\mathbf{x}, \end{aligned} \tag{2.4}$$

with $C(\mathbf{x}, \mathbf{u})$ being the matching cost for each pixel $\mathbf{x}$ of the image, given a depth map estimation $\mathbf{u}^*$. For each pixel $\mathbf{x} = (x, y) \in \Omega$ we search for the minimal matching cost computed over all considered depths $\in [d_{min}, d_{max}]$ and obtain the resulting depth $\mathbf{u}^*$ as the best matching depth for this pixel.
This process is done for each pixel separately in an exhaustive search, leading to noisy results as shown in Figure 2.3(a). To improve the quality of the depth map in areas with weak data terms (no distinguishable texture, moving objects, reflections, ...) we add additional smoothness terms to the energy functional to minimize (see Section 2.3).



(a) Depth map based on minimizing image matching cost only. Census 7x9 was used as cost function - which will be detailed in Section 2.2.

(b) Depth map based on minimizing image matching cost (Census 7x9) plus regularizer (TV) and additional outlier filtering using left-right-check.

Figure 2.3: Reconstruction quality without and with regularizers enforcing local smoothness of surfaces.

Camera models

In this work we assume that the absolute positions and orientations of the cameras as well as their internal parameters are known and optimized w.r.t. each other by e.g., bundle adjusting all input images and their cameras beforehand.
To restrict the search space for image matching from 2D to 1D, we need to establish an epipolar geometry between image pairs. If the cameras can be approximated by the pinhole camera model, the resulting epipolar geometry is mapping one image coordinate in the first image to a corresponding line in the second image. This yields

the usual preprocessing step for stereo reconstruction of rectifying the input images pairwise, such that the epipolar lines are horizontally aligned to the image plane (see e.g., [128],[129],[66],[42]).
In the case of multi-image matching, where the images can be arranged arbitrarily instead of the left-right assumption, this pairwise rectification is cumbersome to implement and introduces further numerical inaccuracies as the rectification homographies apply perspective distortion the images. In general, aligning the epipolar lines for more than 2 (arbitrarily located) images is not feasible anymore.
Furthermore, e.g. satellite images are obtained using a push broom camera (the CCDs are arranged one-dimensional instead of a two-dimensional array) and the corresponding Rational Polynomial Camera (RPC) model (see e.g. [38]) is quite different from the pinhole model. Most notably, the resulting epipolar lines of an image pair are not straight, but curved [77], increasing the complexity of an image rectification approximation.

Pinhole camera model

In this thesis we assume that the lens distortion effects of the input images have already been corrected [129], resulting in the standard projective pinhole camera model for projecting 3D points into 2D image space.

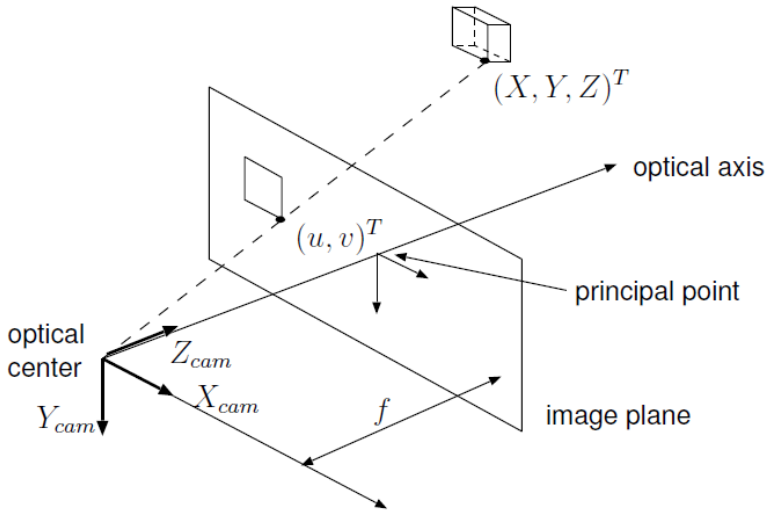


Figure 2.4: Pinhole camera model with the projection of a 3D point $(X, Y, Z)^T$ onto the 2D image plane with coordinates $(u, v)^T$.

$$\tilde{\mathbf{p}}_{2D} = [\mathbf{K}|\mathbf{0}_3] \cdot \mathbf{T}_{world}^{cam} \cdot \tilde{\mathbf{p}}_{3D}$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix}$$

with $\mathbf{T}_{world}^{cam}$ denoting the transformation of the world coordinate origin to the location of the camera center and $\mathbf{K}$ being the intrinsic matrix, containing the calibration parameters from (usually offline) calibration [129],[42]). In case the camera is already placed at the world coordinate origin, the projection of a 3D point into 2D image space reduces to

$$\tilde{\mathbf{p}}_{2D} = [\mathbf{K}|\mathbf{0}_3] \cdot \tilde{\mathbf{p}}_{3D}^{cam}$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$x = \frac{X \cdot f_x}{Z} + p_x$$

$$y = \frac{Y \cdot f_y}{Z} + p_y$$

Push broom camera - RPC model

For an image taken with a push broom camera each image line is taken at a different instance of time (see Figure 2.5). The exterior orientation parameters, i.e. the rotation angles and the position of the perspective center depend on the acquisition time and therefore change from scan line to scan line. The interior orientation parameters, which comprise the focal length, the principal point location, the lens distortion coefficients, and other parameters directly related to the physical design of the sensor, are in general the same for the entire image. A generic push broom camera model can be expressed by modified collinearity equations in which all exterior orientation parameters are defined as a function of time (see e.g. [1], [69], [3], [4]). Nowadays, the RPC model is used for many satellites - only very few have to be modeled by their exact sensor model.
The model for projecting a 3D point j to 2D image space i is given by

$$x_{i,j} = \frac{P_{i1}(X_j, Y_j, Z_j)}{P_{i2}(X_j, Y_j, Z_j)}, \qquad y_{i,j} = \frac{P_{i3}(X_j, Y_j, Z_j)}{P_{i4}(X_j, Y_j, Z_j)}. \tag{2.5}$$

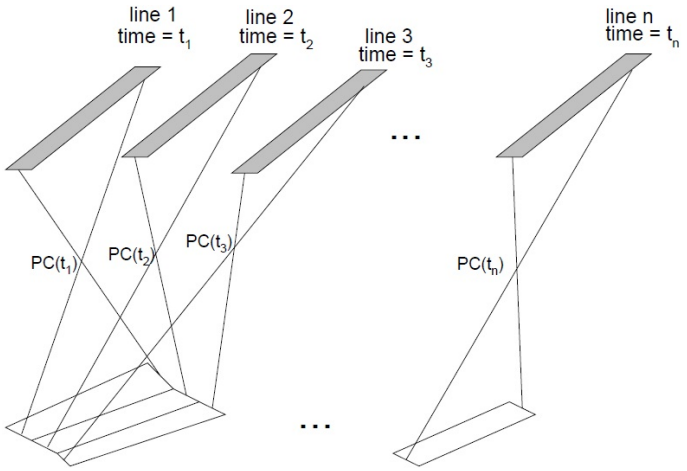


Figure 2.5: Push broom camera model - image taken from [39]. The 2D image is acquired line-wise.

with x_{ij}, y_{ij} being the normalized (offset and scaled) image coordinates and X_j, Y_j, Z_j the corresponding object point coordinates, which refer to normalized latitude, longitude, and altitude. The polynomials are

$$\begin{aligned}
P_{i1}(X,Y,Z) =& (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, \\
& a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}, a_{17}, a_{18}, a_{19}, a_{20}) \cdot \\
& (1, Y, X, Z, YX, YZ, XZ, Y^2, X^2, Z^2, \\
& XYZ, Y^3, YX^2, YZ^2, Y^2X, X^3, XZ^2, Y^2Z, X^2Z, Z^3)^T \\
P_{i2}(X,Y,Z) =& (b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, \\
& b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16}, b_{17}, b_{18}, b_{19}, b_{20}) \cdot \\
& (1, Y, X, Z, YX, YZ, XZ, Y^2, X^2, Z^2, \\
& XYZ, Y^3, YX^2, YZ^2, Y^2X, X^3, XZ^2, Y^2Z, X^2Z, Z^3)^T \\
P_{i3}(X,Y,Z) =& (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, \\
& c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}) \cdot \\
& (1, Y, X, Z, YX, YZ, XZ, Y^2, X^2, Z^2, \\
& XYZ, Y^3, YX^2, YZ^2, Y^2X, X^3, XZ^2, Y^2Z, X^2Z, Z^3)^T \\
P_{i4}(X,Y,Z) =& (d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, \\
& d_{11}, d_{12}, d_{13}, d_{14}, d_{15}, d_{16}, d_{17}, d_{18}, d_{19}, d_{20}) \cdot \\
& (1, Y, X, Z, YX, YZ, XZ, Y^2, X^2, Z^2, \\
& XYZ, Y^3, YX^2, YZ^2, Y^2X, X^3, XZ^2, Y^2Z, X^2Z, Z^3)^T
\end{aligned} \tag{2.6}$$

where $P_{i1}, P_{i2}, P_{i3}, P_{i4}$ are cubic functions in object space coordinates, X, Y, Z are normalized object space coordinates (latitude, longitude, altitude) and x_{ij}, y_{ij} are normalized image space coordinates. For a more detailed description of the RPC model, we refer to the work of [39] and [40].

Unified trilinear interpolation model

Since the camera models can differ a lot in their complexity and their projective functions have to be evaluated numerous times, we pursue a different strategy especially for the satellite based RPC model. We establish the epipolar geometry between two images I_1 and I_2 directly by evaluating the function $F_{(1,2)}(\mathbf{x}, d)$, which projects a pixel $\mathbf{x}$ from I_1 to I_2 using the depth d, for every single pixel of $I_1 \in \Omega$ and every possible depth $d \in D$ individually.
Especially for push broom images and the RPC camera model, evaluation of $F_{(1,2)}(\mathbf{x}, d)$ for every pixel $\mathbf{x}$ and depth hypothesis d is computationally very expensive and cannot be used in practice. We therefore compute $F_{(1,2)}(\mathbf{x}, d)$ only for a sparse (and uniformly distributed) set of grid points in $\Omega \times D$ space. For all other points we interpolate the projected pixel coordinates by using trilinear interpolation (Figure 2.7). The creation of this lookup-table L is done iteratively starting with a very coarse $10 \times 10 \times 10$ grid whose resolution is increased until the reprojection error of the in-between grid points gets smaller than a specified threshold. This allows us to use arbitrary complex camera models while the time for a coordinate transfer $(\mathbf{x}, d) \rightarrow F_{(1,2)}(\mathbf{x}, d)$ still is the one needed for a trilinear interpolation using the lookup table, which can be implemented efficiently.
To furthermore reduce the need for rotational invariant cost functions we apply a plane-sweep approach [23] for computing the image matching cost functions. Given a depth d, we sweep over the reference image I_1, sample image I_2 at the corresponding image position (x', y') and copy the obtained color/intensity to an image $\tilde{I}_2$ at the same position (x, y) as in the reference image. When computing the matching costs of a depth hypotheses d and the whole image I_1, we simply evaluate the cost function at the same position (x, y), using the same local support window in both I_1 and I_2. Note that the whole process runs independently for each pixel of I_1 and therefore can be implemented very efficiently on a GPU.

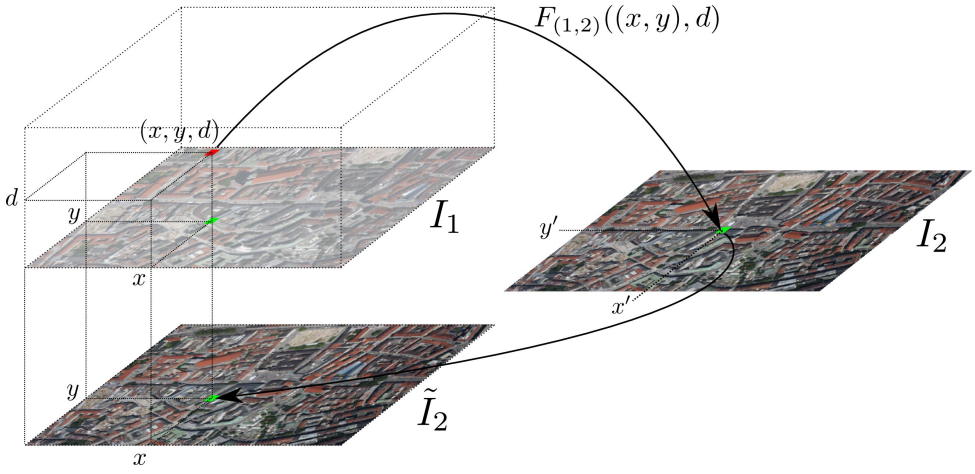


Figure 2.6: For a coordinate (x, y) in image I_1 and the depth d, obtain the corresponding coordinate (x', y') in image I_2 by trilinear interpolation in the sparse lookup table L, sample the pixel color/intensity and copy it to the *warped* image $\tilde{I}_2$ at position (x, y). In short notation: $\tilde{I}_2 = I_2(x', y') = I_2(F_{(1,2)}((x, y), d))$.

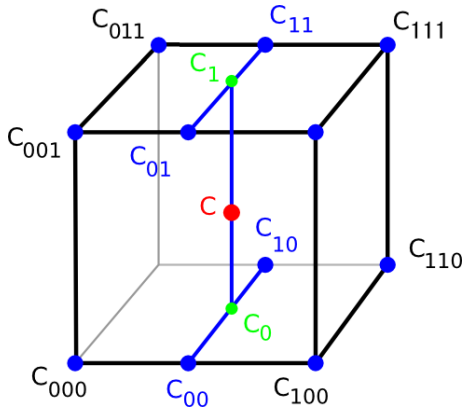
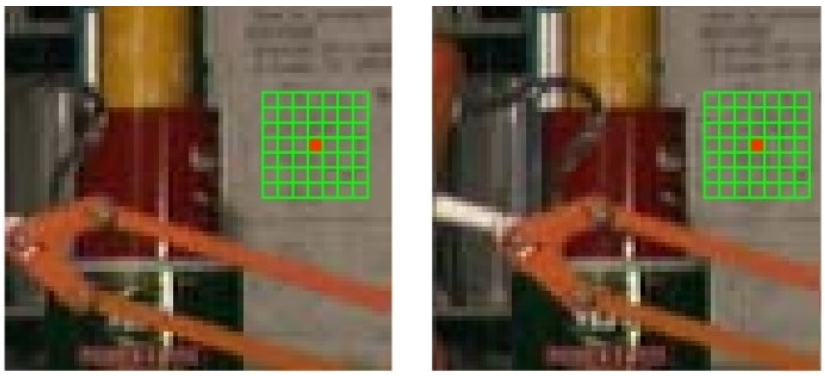


Figure 2.7: Trilinear interpolation

Image Matching Cost Functions

Many dense stereo reconstruction algorithms exhaustively compute a matching quality for every pixel of a reference image and its possible projections in any other image for every depth hypothesis. Computing such dissimilarity measures of two image positions has the challenging task of fulfilling the two competing requirements of a) being as fast as possible and b) as descriptive / disjunctive as possible. Since we perform an exhaustive search over all depths and each pixel, we have to compute

the full cost volume, which for a standard example of 1024×768 images and 128 depths sums up to slightly more than 100 million dissimilarity computations. At the same time, the dissimilarity measure must be descriptive enough to reduce false matches to a minimum.
Due to the speed requirement, typical dissimilarity measures (also called cost function) in dense stereo are: Absolute Differences (AD), Normalized Cross Correlation (NCC), Birchfield-Tomasi measure [8], Mutual Information [116] and Census transform [125]. These are easy to compute, but not very descriptive and therefore prone to false matches.
Of the advanced and more descriptive features (mainly used in image recognition) like SIFT [67], SURF [2], BRIEF [17], BRISK [64] only the DAISY descriptor [114] showed a somewhat acceptable speed for using it in dense stereo reconstruction. Due to some invariance against rotation, scaling and simple radiometric changes, using the DAISY descriptors (or any other of the abovementiond features) yields very good results for wide-baseline stereo, where some simple dissimilarity measures fail because of the large perspective differences. This property makes them good candidates for matching remote sensing stereo images. Unfortunately, as a result of a larger local support window, spatially very closely related pixels have similar (DAISY) descriptors and can't be distinguished very clearly, which results in a blurry reconstruction of sharp depth discontinuities, occurring for example at the sides of high buildings. For more detailed evaluations of different cost functions, see e.g. the surveys in [99], [112], [48], [49]. Due to the nature of our epipolar geometry model and the plane-sweep approach (Section 2.1.2), it is not necessary for the cost function to be rotational invariant. Also the scale invariance can be neglected for standard dense stereo reconstruction as in our case, where all images of a scene are taken from a similar distance.



(a) I_1 with patch (green grid) centered at $\mathbf{x_1}$ (red pixel)

(b) I_2 with patch centered at $\mathbf{x_2}$

Figure 2.8: Image matching cost functions with spatial support window

Normalized Cross Correlation

The normalized cross correlation is basically the prototype for image matching based on spatial support windows W (see Figure 2.8) around the pixel positions to compare and is invariant to additive and multiplicative illumination changes.

$$C_{NCC}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\sum_{[i,j]\in W} \left(I_1(\mathbf{x_1} + [i,j]) - \mu_1(\mathbf{x_1})\right) \cdot \left(I_2(\mathbf{x_2} + [i,j]) - \mu_2(\mathbf{x_2})\right)}{\sigma_1(\mathbf{x_1}) \cdot \sigma_2(\mathbf{x_2})}$$
$$C_{NCC}(\mathbf{x}, d) = C_{NCC}(\mathbf{x}, F_{(1,2)}(\mathbf{x}, d)) \tag{2.7}$$

with μ_i being the mean intensity of the image patch located at the respective positions $\mathbf{x_i}$ and σ_i the standard deviations accordingly.

Modified Census Transform

The Census transform CT as described in [125] is a non-parametric transform which encodes the local image structure within a small patch around a given pixel. It is defined as an ordered set of comparisons of intensity differences and therefore invariant to monotonic transformations which preserve the local pixel intensity order:

$$\xi(I(\mathbf{x}), I(\mathbf{x}')) = \begin{cases} 1 & \text{if } I(\mathbf{x}') < I(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

$$CT(I, \mathbf{x}) = \bigotimes_{[i,j]\in W} \xi(I(\mathbf{x}), I(\mathbf{x} + [i,j])) \ , \tag{2.9}$$

for an ordered set of displacements $W \subset \mathbb{R}^2$ and the operator $\bigotimes$ concatenating the binary values of ξ to a binary string. Image matching is then performed by comparing the resulting binary vectors at different image positions. However, the Census transform strongly depends on the center pixel and a slight variation of its intensity can cause the descriptor to vary significantly. We address this issue by using the following (robustified) modification of the Census transform

$$MCT(I, \mathbf{x}) = \bigotimes_{[i,j]\in W \cup [0,0]} \xi(\bar{I}(\mathbf{x}), I(\mathbf{x} + [i,j])) \ , \tag{2.10}$$

where we replaced the intensity of the center pixel by a weighted average of the intensities in its direct 4-neighborhood (see Figure 2.9). A similar modification is used by [32] for face detection.

The matching cost of different Census vectors s_1, s_2 is then computed as their Hamming distance $d_H(s_1, s_2)$ – number of differing bits – where highest matching quality is achieved for minimal Hamming distance. To simplify further usage of the matching costs, we scale them to the real-valued interval $[0, 1]$ by dividing through

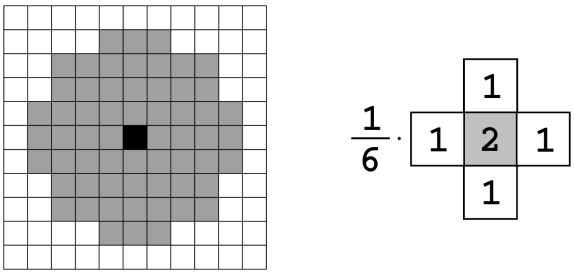


Figure 2.9: Left: The displacement field D used for computing a 61-Bit census transform of the black center pixel. The size of D was chosen deliberately, as to fit in a 64-Bit variable. Right: The weights for computing the center pixel intensity $\bar{I}(\mathbf{x})$.

the maximal cost $\max_{i,j}\{d_H(s_i, s_j)\}$, which equals the number of pixels in the displacement field D.

$$C_{MCT}(\mathbf{x}, d) = \frac{d_H\left(\ MCT(I_1, \mathbf{x}),\ MCT(I_2, F_{(1,2)}(\mathbf{x}, d))\ \right)}{\max_{i,j}\{d_H(s_i, s_j)\}} \tag{2.11}$$

Having a large support window for the Census transform, as shown in Figure 2.9, increases the robustness of the matching function against mismatches, especially when searching through a large range of depths. The support window size of the Census transform is typically 5×5, 7×7, 7×9, or circular spatial structure like Figure 2.9, because due to efficient implementation issues, the resulting binary vectors then fits into 32 or 64 bit variables.
On the other hand, this window-based matching faces several drawbacks: The depth within the window is assumed to be constant and the results therefore are biased towards fronto-parallel surfaces. For the same reason, the resulting depth map gets blurry along discontinuities.
To limit the influence of these drawbacks, the window-based cost function of Equation 2.11 can be combined (weighted sum) with a pixel-wise second cost function using e.g., absolute difference of the intensity values or mutual information.

Mutual Information

Mutual Information [116] has already proven to be a good choice as pixel-wise matching cost function ([51], [44]) and is based on the joint entropy of the involved images. For combination with other cost functions, we normalize the MI cost function to $[0, 1]$

$$C_{MI}(\mathbf{x}, d) = 1 - \frac{\tilde{C}_{MI}(\mathbf{x}, d) - \min_{\mathbf{x},d}\{\tilde{C}_{MI}(\mathbf{x}, d)\}}{\max_{\mathbf{x},d}\{\tilde{C}_{MI}(\mathbf{x}, d)\} - \min_{\mathbf{x},d}\{\tilde{C}_{MI}(\mathbf{x}, d)\}} \tag{2.12}$$

with

$$\tilde{C}_{MI}(\mathbf{x}, d) = mi_{I_1,I_2}(\ I_1(\mathbf{x}),\ I_2(F_{(1,2)}(\mathbf{x}, d))\ ) \tag{2.13}$$

and mi_{I_1,I_2} being the mutual information according to [116]. As Equation 2.13 requires knowledge of the depth d a priori, a hierarchical approach is used to get a good estimate for $\tilde{C}_{MI}$ (see [44]).

Adaptive Support Weights

Window-based image matching suffers from the "foreground fattening" phenomenon when support windows are located on depth discontinuities, such as partially covering a roof top and the adjacent street. To limit this effect, we locally aggregate the *raw* image matching $cost(\mathbf{x_1}, \mathbf{x_2})$ of two pixel locations $\mathbf{x_1}$ and $\mathbf{x_2}$ – e.g., from Equation 2.7, 2.11, 2.12 – using adaptive support-weights [124] for corresponding pixels $\mathbf{p}$ in I_1 and $\mathbf{q}$ in I_2:

$$C(\mathbf{p}, \mathbf{q}) = \frac{\sum_{\tilde{\mathbf{p}} \in N_{\mathbf{p}}, \tilde{\mathbf{q}} \in N_{\mathbf{q}}} [w(\mathbf{p}, \tilde{\mathbf{p}}) \cdot w(\mathbf{q}, \tilde{\mathbf{q}}) \cdot cost(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})]}{\sum_{\tilde{\mathbf{p}} \in N_{\mathbf{p}}, \tilde{\mathbf{q}} \in N_{\mathbf{q}}} [w(\mathbf{p}, \tilde{\mathbf{p}}) \cdot w(\mathbf{q}, \tilde{\mathbf{q}})]} \tag{2.14}$$

The weights $w(\mathbf{p}, \mathbf{q})$ are based on color differences $\Delta_{col}(\mathbf{p}, \mathbf{q})$ and spatial distances $\Delta_{dist}(\mathbf{p}, \mathbf{q})$,

$$w(\mathbf{p}, \mathbf{q}) = \exp\left(-\frac{\Delta_{col}(\mathbf{p}, \mathbf{q})}{\gamma_{col}} - \frac{\Delta_{dist}(\mathbf{p}, \mathbf{q})}{\gamma_{dist}}\right), \tag{2.15}$$

which are open for tuning but usually reside in the range of $\gamma_{dist} = 4$ (= radius of the support window) and $\gamma_{col} = 5.0$ for 8-bit images ($\gamma_{col} = 20.0$ for 11-bit images respectively). As this local aggregation favors fronto-parallel surfaces, we keep this radius relatively small (4 pixel), to keep a balance between increased accuracy along discontinuities and not "over-favoring" fronto-parallel surfaces.

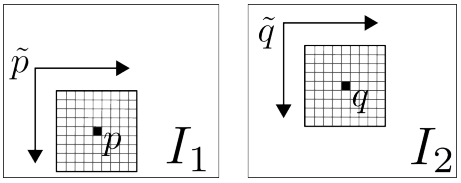
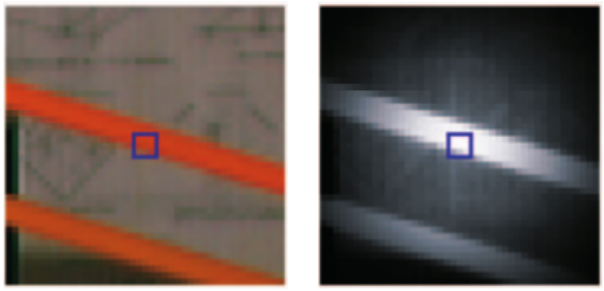


Figure 2.10: Basic scheme for the adaptive support-weights as described in Equation 2.14. In a local neighborhood around the considered matching pixel positions $\mathbf{p}$ and $\mathbf{q}$, the matching cost of all possible matching coordinates are weighted and summed up, according to the scheme described in this Section 2.2.4.

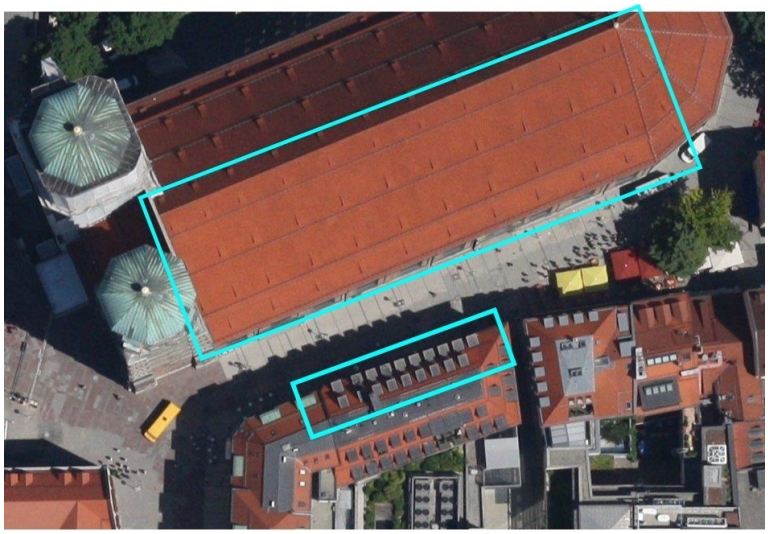


(a) (b)

Figure 2.11: (a) Input image with center pixel marked by rectangle. (b) Corresponding support-weights incorporating both color and spatial distance (brighter pixel corresponding to larger support-weights). Image taken from [124].

Shortcomings

As much effort as one might put into image matching cost functions, there are many cases where image matching is basically not possible - see Figure 2.12 for some examples. The so-called data term is too weak, misleading or ambiguous for reliable estimation of the true optimal solution using image matching alone (see Figure 2.13). Two overcome this problem an additional regularization term is used for reconstruction of piecewise smooth solutions - which will be described in the following section.

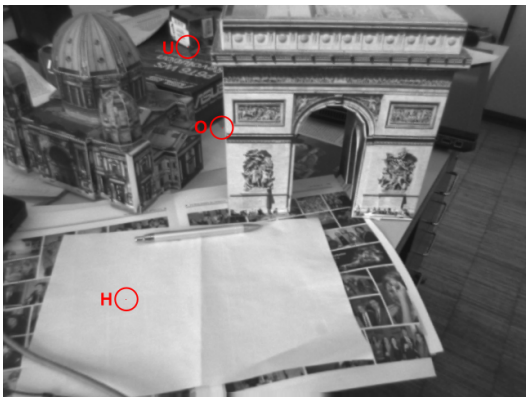


(a) Repetitive texture along the shingles of the roofs

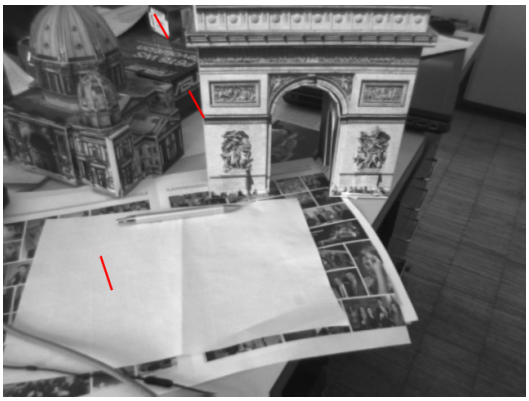
(b) Textureless regions along the flat roof

(c) Sensor oversaturation and specular reflections

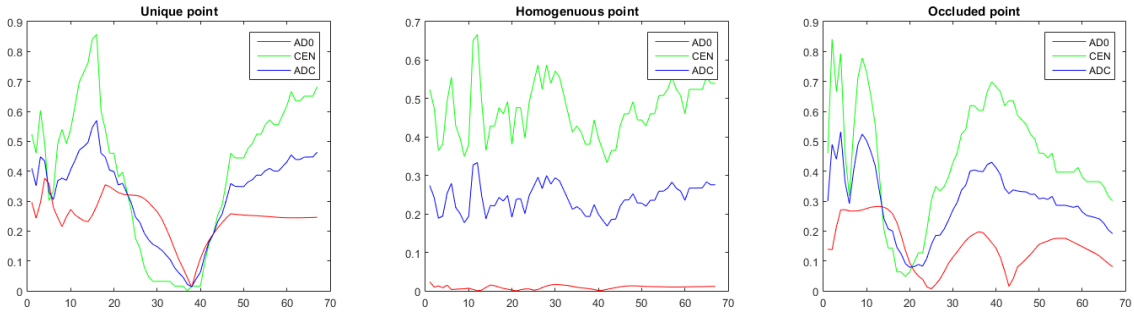
Figure 2.12: Examples of image regions ill-suited for image matching



(a) Reference image with selected points: unique (U), homogeneous (H), occluded (O).



(b) Second matching image with search space along epipolar lines.



(c) Image matching cost functions for each of the three considered points (U,H,O): pixelwise absolute difference (AD0), Census 5×5 (CEN), and weighted mean of both (ADC).

Figure 2.13: Image matching cost functions for selected points and their 3D search space along the corresponding epipolar lines. For details on improving the quality of the data term via multiple measurements see Chapter 4.

Total Variation Regularization

Since the raw image matching costs are still prone to mismatches and noise, it is a bad choice to solve for the depth map pixel-wise by choosing the depth with minimum matching cost (compare Figure 2.3(a)). Adding additional smoothness priors is a well established technique to mitigate the effect of erroneous data terms, forcing the depth map to be locally smooth.

$$\begin{aligned} \mathbf{u}^* &= \arg\min_{\mathbf{u}} \left\{ \int_\Omega E_{data}(\mathbf{u}(\mathbf{x})) + E_{smooth}(\mathbf{u}(\mathbf{x})) \, \mathrm{d}\mathbf{x} \right\} \\ &= \arg\min_{\mathbf{u}} \left\{ \int_\Omega C(\mathbf{u}(\mathbf{x})) + \lambda \cdot h(\mathbf{u}(\mathbf{x})) \, \mathrm{d}\mathbf{x} \right\} \end{aligned} \tag{2.16}$$

The data term C still measures the quality of the matching image patches and is now balanced against a smoothing functional h with a controllable scalar weighting factor λ. Compared to the pixelwise solution of Equation 2.4, this energy is non-trivial to solve, since the smoothness constraints (implied by the smoothing function h) are typically based on first- or second-order derivatives of the depth map and therefore cannot be optimized pixelwise anymore. The choice of the data term and smoothness energy functional are the most important issues, since they both affect the property of preserving the original signal as well as being able to solve the resulting optimization problem accurately and efficiently. Typically, the data term is not convex in the variable $\mathbf{u}$ to solve for, while the regularization term implies difficulties for an efficient optimization scheme.

Total Variation

We first give a formal definition of the total variation and its higher order counterpart the total generalized variation and afterwards illustrate their properties in an example of denoising 2D data.

Definition 1 C^k functions
Let k be a non-negative integer. The function f is said to be of class C^k if the derivatives $f', f'', ..., f^{(k)}$ exist and are continuous. The function f is said to be of class C^∞, or smooth, if it has derivatives of all orders.

Definition 2 L^p space
The space of p-integrable functions is defined as

$$\mathcal{L}^p(\Omega) := \left\{ f : \Omega \to \mathbb{R} \mid \left(\int_\Omega |f|^p \, \mathrm{d}x \right)^{1/p} < \infty \right\} \tag{2.17}$$

Definition 3 Sobolev space $W^{k,p}$
The Sobolev space $W^{k,p}(\Omega)$ is defined to be the set of all functions $u \in L^p(\Omega)$ such that for every n-tuple α with $|\alpha| \leq k$, the weak partial derivative $D^\alpha u$ belongs to $L^p(\Omega)$

$$W^{k,p}(\Omega) = \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega)\ \forall |\alpha| \leq k\}. \tag{2.18}$$

Intuitively, a Sobolev space is a space of functions with sufficiently many derivatives for e.g., partial differential equations and is equipped with a norm that measures both the size and regularity of a function.

Definition 4 Weak partial derivative
Given an open set $\Omega \subset \mathbb{R}^n$, a function $f \in L^1$ is weakly differentiable with respect to x_i if there exists a function $g_i \in L^1$ such that

$$\int_\Omega f \partial_i \phi \, \mathrm{d}x = -\int_\Omega g_i \phi \, \mathrm{d}x, \tag{2.19}$$

for all functions ϕ being infinitely differentiable and with compact support in Ω, i.e. $\phi \in C_c^\infty(\Omega)$. The function g_i is called the weak ith partial derivative of f, and is denoted by $\partial_i f$.
Weak derivatives generalize the concept of the (strong) derivative of a function for functions which are not differentiable, but only integrable. The main idea behind the weak derivative is that Equation 2.19 allows to shift the differential operator from one variable to another one which is defined to be always differentiable.

Definition 5 Total Variation
Given a function $u \in L^1(\Omega)$ on a bounded domain $\Omega \subset \mathbb{R}^n$ with $n \geq 2$, the total variation of u is defined as

$$\mathrm{TV}(u) := \sup \left\{ \int_\Omega u(x) \cdot \mathrm{div}(\phi(x)) \, \mathrm{d}x \ : \ \phi \in C_c^1(\Omega, \mathbb{R}^n),\ ||\phi||_\infty \leq 1 \right\} \tag{2.20}$$

with

$$\mathrm{div}(\phi(x)) = \sum_{i=1}^{n} \frac{\partial \phi_i}{\partial x_i}(x) \tag{2.21}$$

and if $u \in W^{1,1}(\Omega)$, see e.g. [21], the total variation of u can be written as

$$\mathrm{TV}(u) = \int_\Omega |\nabla u|_2 \, \mathrm{d}x. \tag{2.22}$$

Definition 6 Total Generalized Variation
The total generalized variation of order k with weights α is defined as

$$\mathrm{TGV}_{\alpha}^{k}(u) := \sup \left\{ \int_{\Omega} u \cdot \mathrm{div}^{k} \phi \, \mathrm{d}x \; : \phi \in C_{c}^{k}(\Omega, \mathrm{Sym}^{k}(\mathbb{R}^{n})), \right. \\ \left. ||\mathrm{div}^{l} \phi||_{\infty} \leq \alpha_{l}, \;\; l = 0, ..., k-1 \right\} \tag{2.23}$$

with $k \geq 1$ and $\alpha_0, ..., \alpha_{k-1} \geq 0$. $\mathrm{Sym}^{k}(\mathbb{R}^{n})$ denotes the space of symmetric tensors of order k with arguments in $\mathbb{R}^{n}$. Note that for $k = 1$ and $\alpha > 0$ it holds that

$$\begin{aligned} \mathrm{TGV}_{\alpha}^{1}(u) &= \sup \left\{ \int_{\Omega} u \cdot \mathrm{div} \; \phi \, \mathrm{d}x \; : \; \phi \in C^{1}(\Omega, \mathrm{Sym}^{1}(\mathbb{R}^{n})), ||\phi||_{\infty} \leq \alpha \right\} \\ &= \alpha \cdot \mathrm{TV}(u) \end{aligned} \tag{2.24}$$

implying that TGV is indeed a generalization of TV.

Depth map denoising example

Due to its simplicity we use the application of depth map denoising to illustrate the effects of total variation regularization as well as different norms for the data term. Given a corrupted 2D depth map $\mathbf{f}$, we want to obtain denoised 2D data $\mathbf{u}$.

Tikhonov model

The quadratic model (or Tikhonov model [113]) is one of the earliest and simplest regularization methods used for ill-posed problems. It is defined as the quadratic variational problem

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} (\mathbf{u} - \mathbf{f})^{2} \, \mathrm{d}x \; + \; \lambda \int_{\Omega} |\nabla \mathbf{u}|_{2}^{2} \, \mathrm{d}x \right\} \;\; , \tag{2.25}$$

The quadratic model tries to find a smooth solution u which minimizes the squared distance to the observations f. Being quadratic in u, the Tikhonov model poses a simple optimization problem, but it leads to an oversmoothing of edges and the quadratic data term is not robust against strong outliers in the observed data. For this reason, we do not consider this model at all and only mention it here for sake of completeness.

ROF model

The seminal paper of [96] introduced the Rudin-Osher-Fatemi model (ROF-model) as an edge preserving 2D image restoration model by applying total variation as

regularization term

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} (\mathbf{u} - \mathbf{f})^2 \, \mathrm{d}x \; + \; \lambda \int_{\Omega} |\nabla \mathbf{u}|_2 \, \mathrm{d}x \right\} \quad . \tag{2.26}$$

Note that the ROF model is convex and has a unique global minimizer.

TV-L_1 model

By substituting the quadratic data term in the ROF model with it's L_1 pendant we arrive at the TV-L_1 model

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} |\mathbf{u} - \mathbf{f}| \, \mathrm{d}x \; + \; \lambda \int_{\Omega} |\nabla \mathbf{u}|_2 \, \mathrm{d}x \right\} \quad . \tag{2.27}$$

The difference to the ROF model is that discontinuities in the data are well preserved, since deviations in the data term are not penalized quadratically anymore, but only linearly. This makes the TV-L_1 model much more robust to outliers in the data term. One can say that the ROF model is a good choice if the assumed noise is of Gaussian nature and the TV-L_1 model should be used if white noise (outliers) are present in the data. The TV-L_1 model unfortunately is not strictly convex anymore and does not have a unique global minimizer.

TV-Huber model

The advantage of choosing the L_1 norm for $E_{smooth} = |\nabla \mathbf{u}|_1$ however does not come without cost. As shown in Figure 2.14, minimizing the Total Variation leads to staircasing effects on otherwise smooth data in the resulting reconstruction. To overcome this issue a slight modification of the regularization functional by replacing the L_1 norm with the Huber loss (see Figure 2.15 and Equation 2.29 below) was used by [120] to regularize optical flow estimation.

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} |\mathbf{u} - \mathbf{f}| \, \mathrm{d}x \; + \; \lambda \int_{\Omega} |\nabla \mathbf{u}|_h \, \mathrm{d}x \right\} \tag{2.28}$$

The Huber loss still preserves the main advantage of the L_1 norm for penalizing deviations in the first order derivative only linearly, however small deviations are now penalized quadratically (see Equation 2.29) leading to smooth surfaces and mitigating the stair casing effect.

Definition 7 Huber loss
The Huber loss is a hybrid based on the L_1/L_2 -norm which is robust to outliers

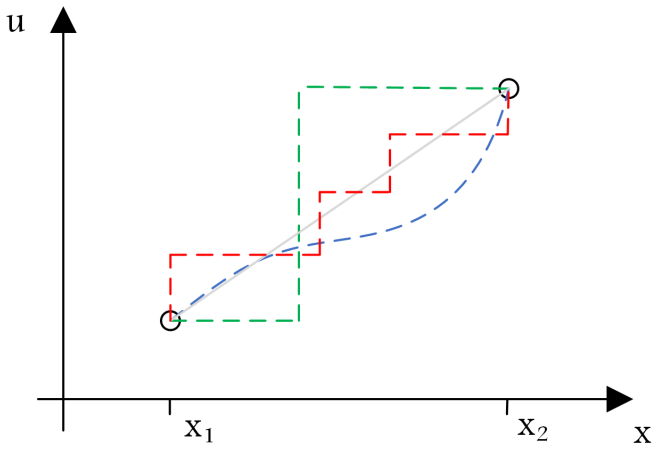


Figure 2.14: Staircasing effect of Total Variation. Without any additional data **u** between the positions x_1 and x_2, regularization via minimization Total Variation does not have a unique solution as every dashed line has the same amount of total variation in u. The gray line corresponds to a smooth regularization between the data points.

(quadratic penalties as in L^2-norm in a local environment and linear penalties L_1 for outliers). It is defined as

$$|x|_h = \begin{cases} \frac{|x|^2}{2h} & \text{if } |x| \leq h \\ |x| - \frac{h}{2} & \text{if } |x| > h \end{cases} \tag{2.29}$$

Furthermore, the Huber loss is fully differentiable with

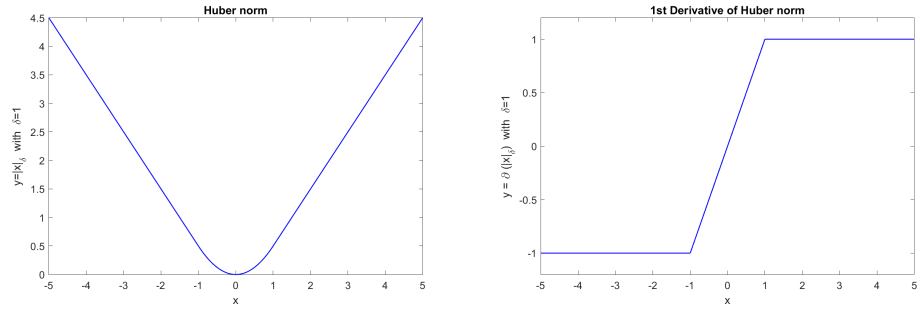


Figure 2.15: Huber loss $|x|_h$ with $h = 1.0$

$$\frac{\partial |x|_h}{\partial x} = \begin{cases} 2|x| \frac{x}{|x|} \frac{1}{2h} & \text{if } |x| < h \\ \frac{x}{|x|} & \text{else} \end{cases} = \begin{cases} \frac{x}{h} & \text{if } |x| < h \\ sgn(x) & \text{else} \end{cases} \tag{2.30}$$

TGV-L_1 model

The Huber loss still is not a completely satisfactory solution for favoring smooth non fronto-parallel surfaces in the minimization process as it implies the inliers of the data to be contaminated by Gaussian noise. Instead, if we take the TV-L_1 model and replace the first order smoothness constraint with a second order smoothness constraint, we obtain the second order Total Generalized Variation [14], favoring any affine surface in the image

$$TGV_2^{\alpha}(\mathbf{u}) = \min_{\mathbf{u},\mathbf{v}} \left\{ \int_{\Omega} |\mathbf{u} - \mathbf{f}| \, \mathrm{d}x \; + \; \alpha_1 \int_{\Omega} |\nabla \mathbf{u} - \mathbf{v}| \, \mathrm{d}x \; + \; \alpha_0 \int_{\Omega} |\nabla \mathbf{v}| \mathrm{d}x \right\} \quad (2.31)$$

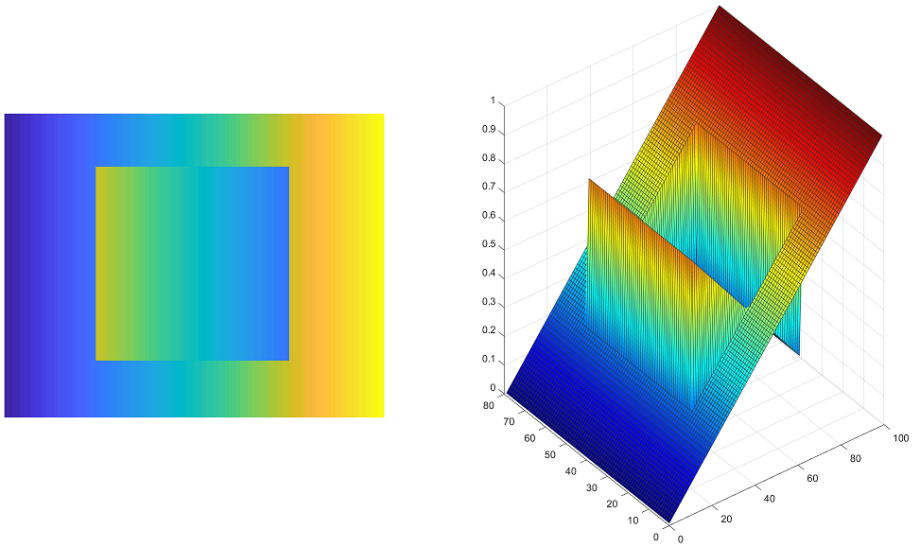
Intuitively, before the variation of the image $\mathbf{u}$ is measured, a 2D vector field $\mathbf{v}$ is subtracted from the gradient of $\mathbf{u}$. An affine surface in the image $\mathbf{u}$ has a constant gradient $\nabla \mathbf{u}$, so by coupling and minimizing $|\nabla \mathbf{u} - \mathbf{v}|$, the vector field $\mathbf{v}$ will also be constant and it's gradient $\nabla \mathbf{v}$ therefore zero. Regarding our overall optimization problem, this means that the energy term will be lower, if affine functions can be found in the image, whereas non-affine functions get additional penalties by $|\nabla \mathbf{v}|$.

Effects of regularization models

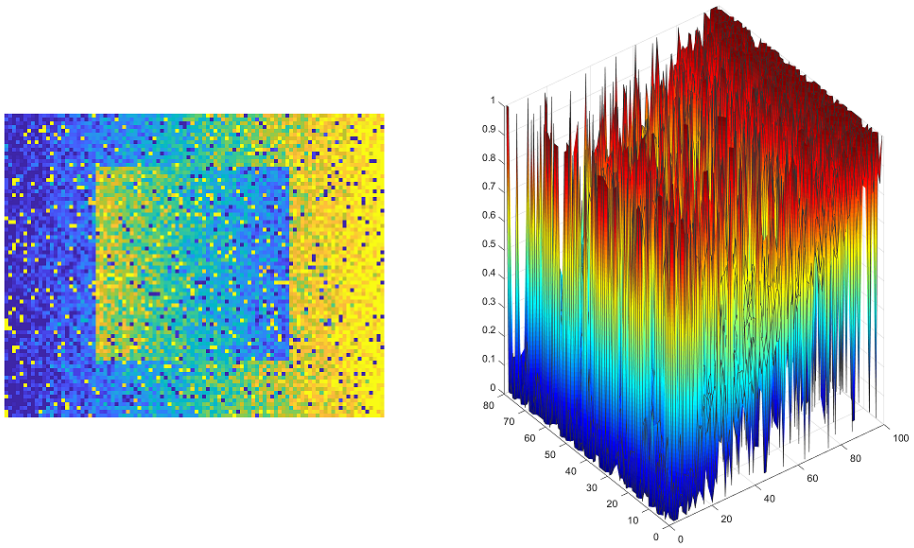
Having given an overview of different energy functionals for regularization, we illustrate their respectively discussed effects by reconstructing a noisy depth map. For a given depth map depicted in image 2.16, we added Gaussian and White noise and solved the energy functionals stated above. The resulting restored surfaces are depicted in Figure 2.17 and 2.18 together with the corresponding PSNR (peak signal-to-noise ration) defined as

$$PSNR(\mathbf{u}, \mathbf{f}) = 10 \cdot \log_{10} \left(\frac{\max(\mathbf{u})^2}{MSE(\mathbf{u}, \mathbf{f})} \right) \quad (2.32)$$

with MSE being the mean squared error between $\mathbf{u}$ and $\mathbf{f}$ and the maximum data value being 1.0 as we normalize the input data to the interval $[0, 1]$.

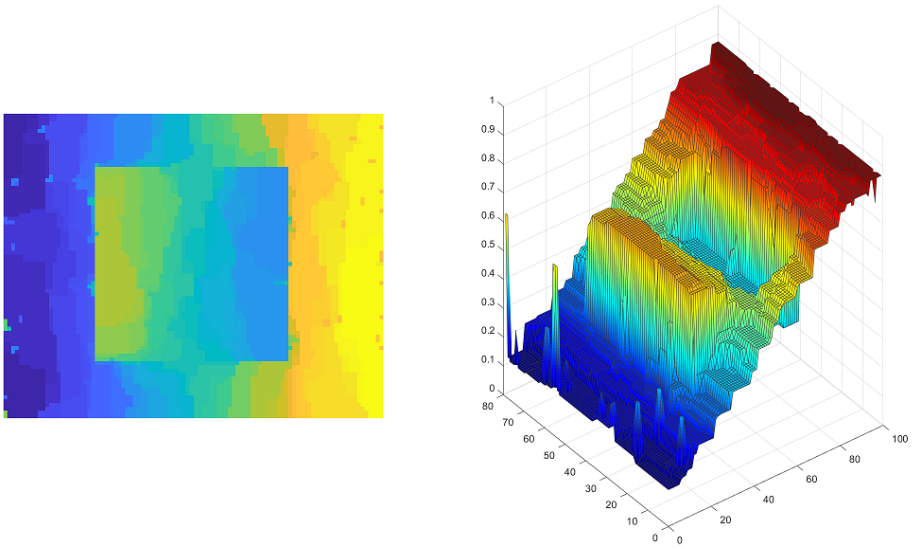


(a) Original data

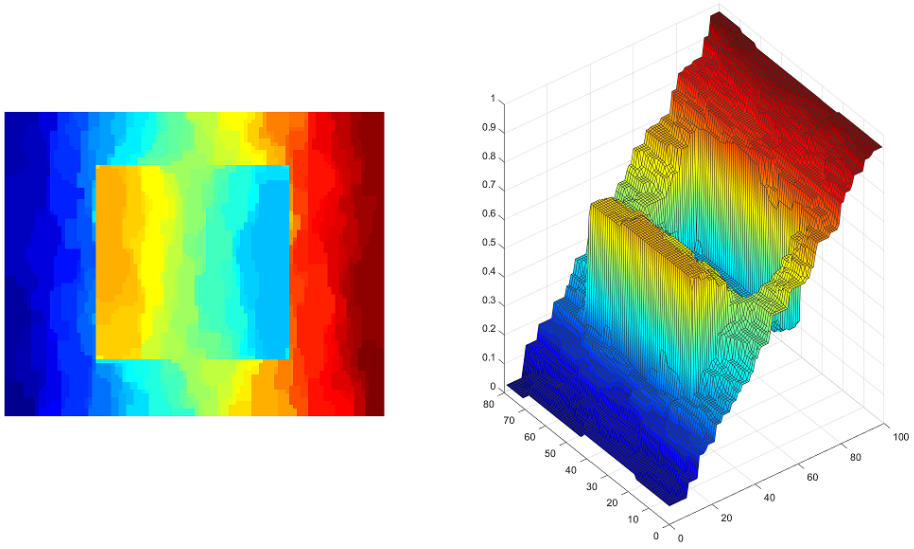


(b) Corrupted data (Gaussian noise and white noise)

Figure 2.16: Impact of different regularization energy functionals on the reconstruction of a noisy 2.5D surface / depth map: Original data and noisy input.

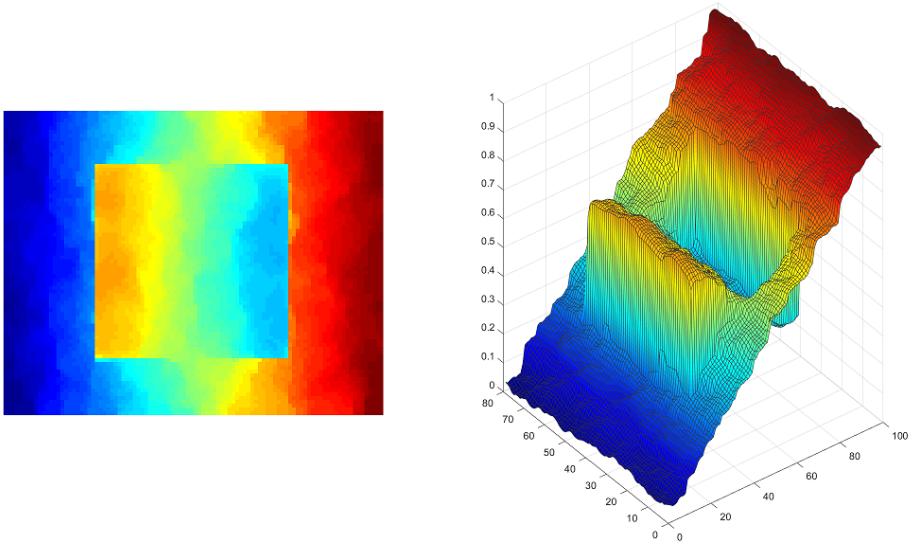


(a) ROF model (PSNR 23.61)

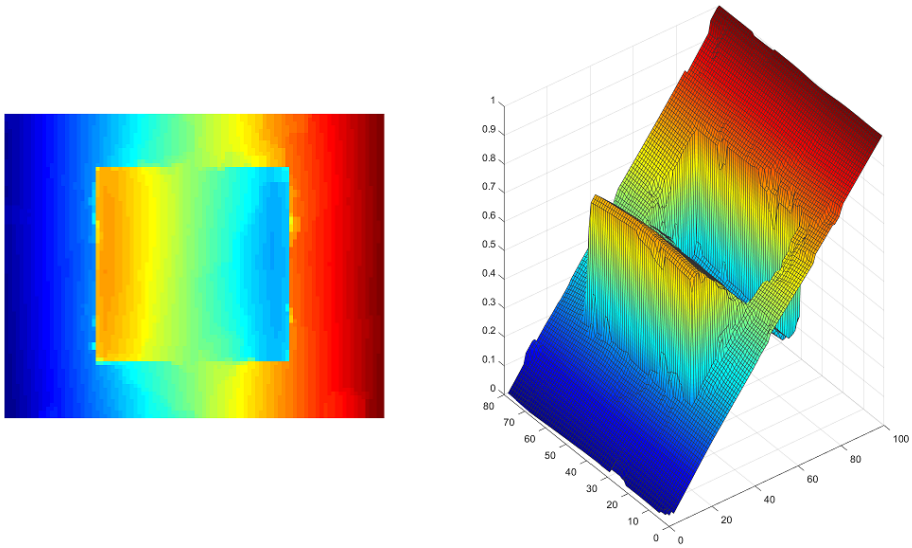


(b) TV-L_1 model (PSNR 31.24)

Figure 2.17: Impact of different regularization energy functionals on the reconstruction of a noisy 2.5D surface / depth map: Resulting reconstruction for ROF-model and TV-L_1 model.



(a) TV-Huber model (PSNR 32.40)



(b) TGV-L_1 model (PSNR 34.71)

Figure 2.18: Impact of different regularization energy functionals on the reconstruction of a noisy 2.5D surface / depth map: Resulting reconstruction for TV-Huber model and TGV-L_1 model.

Optimization

In the following we will describe the underlying numerical optimization procedure for solving (minimizing) the energy functionals stated in this chapter so far. The primal-dual algorithm of Chambolle and Pock [22] is the main optimization framework in this work and will be introduced in the following. To account for better readability we first state the motivation and derivation of the primal-dual algorithm and give the necessary mathematical background definitions afterwards.

Primal-Dual Algorithm

In Equation 2.16 we introduced the general form of energy functionals we want to minimize

$$\mathbf{u}(\mathbf{x}) = \arg\min_{\mathbf{u}} \left\{ \int_{\Omega} E_{data}(\mathbf{u}(\mathbf{x})) + E_{smooth}(\mathbf{u}(\mathbf{x})) \, \mathrm{d}\mathbf{x} \right\} \quad , \tag{2.33}$$

which can be transformed into a general class of energy minimization problems which are well-investigated for computer vision tasks:

$$\min_{\mathbf{x} \in X} \{F(K\mathbf{x}) + G(\mathbf{x})\} \tag{2.34}$$

with F and G being proper, lower semi-continuous convex functions and a linear operator $K \in \mathbb{R}^{n \times m}$. Usually, $F(K\mathbf{x})$ corresponds to the regularization term of the form $||K\mathbf{x}||$ and $G(\mathbf{x})$ to the data term. However, due to the L_1 norm of the total variation and it's non-smoothness, it is difficult to minimize this non-linear problem directly. To overcome this problem [22] transformed the original problem (minimization with respect to a primal variable) into a primal-dual saddle-point problem. Additionally, the algorithm works iteratively by computing local solutions and distributing this information across neighboring image pixels, thus making the primal-dual algorithm highly parallelizable and well suited for implementation on modern GPUs.
The saddle-point formulation of Equation 2.34 is now derived by substituting $F(K\mathbf{x})$ with it's convex conjugate, which by definition of the Legendre-Fenchel transform and F being a convex function is

$$F(K\mathbf{x}) = \max_{\mathbf{y} \in Y} \{\langle K\mathbf{x}, \mathbf{y} \rangle - F^*(\mathbf{y})\} \tag{2.35}$$

yielding

$$\min_{\mathbf{x} \in X} \max_{\mathbf{y} \in Y} \{\langle K\mathbf{x}, \mathbf{y} \rangle - F^*(\mathbf{y}) + G(\mathbf{x})\} \quad . \tag{2.36}$$

Because the dot product is commutative, we can write

$$\langle K\mathbf{x}, \mathbf{y}\rangle = \left\langle \mathbf{x}, K^T\mathbf{y}\right\rangle \tag{2.37}$$

which gives us the final primal-dual formulation (generic saddle-point problem), used in the following optimization problem

$$\min_{\mathbf{x}\in X}\max_{\mathbf{y}\in Y}\{\left\langle \mathbf{x}, K^T\mathbf{y}\right\rangle - F^*(\mathbf{y}) + G(\mathbf{x})\} \tag{2.38}$$

Under the weak assumptions in convex analysis, min and max can be switched in Equation 2.38 which now describes the algorithm:
Perform an alternating gradient descent in the primal variable and a gradient ascent in the dual variable, each step followed by a projection of the intermediate solution onto the constrained set of allowed solutions.
The formal primal-dual algorithm proposed by [22] is then given as

$$\mathbf{y}^{n+1} = \mathrm{Prox}_{\sigma F^*}(\mathbf{y}^n + \sigma K\overline{\mathbf{x}}^n) \tag{2.39}$$
$$\mathbf{x}^{n+1} = \mathrm{Prox}_{\tau G}(\mathbf{x}^n - \tau K^*\mathbf{y}^{n+1}) \tag{2.40}$$
$$\overline{\mathbf{x}}^{n+1} = \mathbf{x}^{n+1} + \theta(\mathbf{x}^{n+1} - \mathbf{x}^n) \tag{2.41}$$

with $\mathrm{Prox}_{\sigma F^*}$ and $\mathrm{Prox}_{\tau G}$ being the proximal mappings onto the constrained sets of $\mathbf{x}$ and $\mathbf{y}$ respectively and initial solutions for $\overline{\mathbf{x}}^0$ and $\mathbf{y}^0$. The third line of the algorithm corresponds to a linear extrapolation step (typically $\theta = 1.0$) and can be seen as an approximate extragradient step, speeding up the convergence. For $0 \leq \theta \leq 1$ and step sizes bound to the operator norm by $\sigma \cdot \tau \cdot \|K\|^2 < 1$, $\mathbf{x}^n$ converges to a minimizer of the original energy function.

Legendre-Fenchel Transformation

The main principle of the primal-dual approach is duality - the principle of looking at a function or problem from two different perspectives (the primal and dual form). Like the Fourier Transform, the Legendre-Fenchel transformation is one mapping of functions $f(\mathbf{x})$ to another space where these functions can be analyzed better and more easy. It maps the $(\mathbf{x}, f(\mathbf{x}))$ space to the space of slope and conjugate $(\mathbf{p}, f^*(\mathbf{p}))$. The Legendre-Fenchel transformation of a continuous but not necessarily differentiable function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, is defined as

$$f^*(\mathbf{p}) = \sup_{\mathbf{x}\in\mathbb{R}^n}\{\mathbf{x}^T\mathbf{p} - f(\mathbf{x})\}\ . \tag{2.42}$$

where $\mathbf{p}$ is the slope and $f^*(\mathbf{p})$ is called the convex conjugate of the function $f(\mathbf{x})$. The conjugate allows building a dual problem which may be easier to solve than

the primal problem. Also note that the Legendre-Fenchel conjugate is always convex. The definition of the Legendre-Fenchel transformation can be interpreted as an encoding of the convex hull of the function's epigraph in terms of its supporting hyperplanes (see Figure 2.19). A closed convex set is uniquely defined by it's supporting hyperplanes.

Definition 8 Supporting hyperplane
A supporting hyperplane of a set S in Euclidean space $\mathbb{R}^n$ is a hyperplane where the set S is entirely contained in one of the two closed half-spaces bounded by the hyperplane and S has at least one boundary-point on the hyperplane.

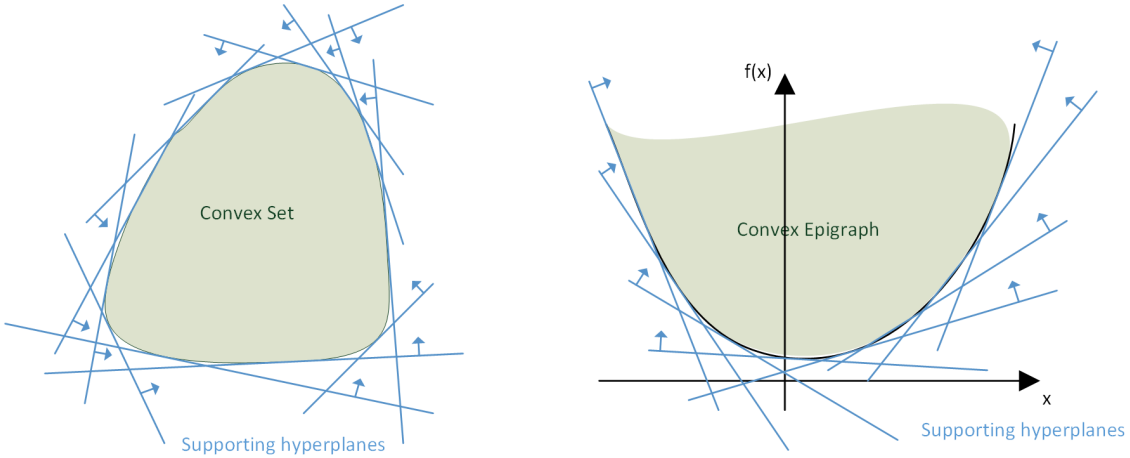


Figure 2.19: A closed convex set is uniquely defined by it's supporting hyperplanes and a convex function is uniquely defined by it's lower supporting hyperplanes.

Example $f(\mathbf{x}) = a \cdot \|\mathbf{x}\|$

The Legendre-Fenchel transform of the L_1 or L_2 norm is given as

$$f^*(\mathbf{p}) = \sup_{\mathbf{x}\in\mathbb{R}^n} \{\mathbf{x}^T\mathbf{p} - f(\mathbf{x})\} \tag{2.43}$$

$$= I_{\{\|\mathbf{p}\|_2 \le a\}}(\mathbf{p}) = I_P(\mathbf{p}) = \begin{cases} 0 & \text{if } \|\mathbf{p}\|_2 \le a \\ \infty & \text{otherwise} \end{cases} \tag{2.44}$$

where $I_{\{\|\mathbf{p}\|_2 \le a\}}(\mathbf{p}) = I_P(\mathbf{p})$ is the so-called *indicator function* on the set $P := \{\mathbf{p} \in \mathbb{R}^n \mid \|\mathbf{p}\|_2 \le a\}$.

$$f(\mathbf{x}) = \|\mathbf{x}\|_\delta$$

For the Huber loss defined as

$$f(\mathbf{x}) = \|\mathbf{x}\|_h = \begin{cases} \frac{|\mathbf{x}|_2^2}{2h} & \text{if } |\mathbf{x}|_2 \leq h \\ |\mathbf{x}|_2 - \frac{h}{2} & \text{if } |\mathbf{x}|_2 > h \end{cases} \tag{2.45}$$

the corresponding Legendre-Fenchel transform computes as

$$f^*(\mathbf{p}) = \begin{cases} \frac{h}{2}\|\mathbf{p}\|_2^2 & \text{if } \|\mathbf{p}\|_2 \leq h \\ \frac{h}{2} & \text{if } h \leq \|\mathbf{p}\|_2 \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{2.46}$$

$$= I_{\|\mathbf{p}\|_2 \leq 1} + \frac{h}{2}\|\mathbf{p}\|_2^2 \tag{2.47}$$

Proximal mapping

For the primal-dual algorithm to be of practical use, an efficient computation of the proximal mapping of F and G is required. In the primal-dual algorithm, we perform a gradient descent step in the primal variable and then need to perform the proximal mapping, to enforce the implied constraints. For the following gradient ascent step in the dual variable we do likewise.
The proximal mapping of a convex function f is defined as

$$\text{prox}_{\sigma f}(\mathbf{x}) = \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + \sigma f(\mathbf{y}) \right\} \quad , \tag{2.48}$$

with $\sigma \in \mathbb{R}$ a scalar factor usually denoting a step-size in gradient descent/ascent algorithms. In literature, the terms *proximal mapping*, *proximity operator*, *prox operator* and *resolvent operator* are used synonymously.

Example $f(\mathbf{x}) = I_C(\mathbf{x})$

The proximal mapping of the indicator function of a convex set C is the projection (shortest distance) on the set C

$$\text{prox}_f(\mathbf{x}) = \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + f(\mathbf{y}) \right\} \tag{2.49}$$

$$= \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + I_C(\mathbf{y}) \right\} \tag{2.50}$$

$$= \arg\min_{\mathbf{y} \in C} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \right\} \tag{2.51}$$

$$= \Pi_C(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \mathbf{x} \in C \\ proj_C(\mathbf{x}) & \text{if } \mathbf{x} \notin C \end{cases} \tag{2.52}$$

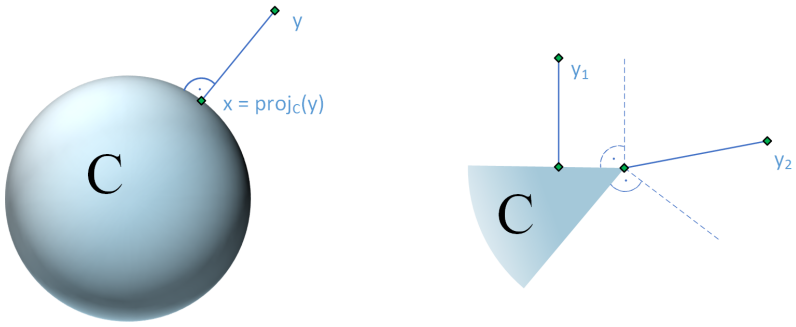


Figure 2.20: Examples for proximal mapping. In case of n-dimensional unitballs the prox operator is a Euclidean projection (left). In general each point is projected along the shortest line onto the convex set C.

In the derivation above we used the fact that $I_C(y) = \infty$ for all $y \notin C$ and $I_C(y) = 0$ for all $y \in C$.

For example, if we have the *unitball* constraint

$$C = \{\mathbf{y} \mid \|\mathbf{y}\| \leq 1\} \tag{2.53}$$

then

$$\text{prox}_{I_C}(\mathbf{x}) = \Pi_C(\mathbf{x}) = \frac{\mathbf{x}}{\max(1, \|\mathbf{x}\|_2)} \tag{2.54}$$

Example $f(\mathbf{x}) = \frac{\lambda}{2} \cdot \|\mathbf{x} - \mathbf{g}\|_2^2$

$$\begin{aligned} \text{prox}_f(\mathbf{x}) &= \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + f(\mathbf{y}) \right\} \\ &= \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{y} - \mathbf{g}\|_2^2 \right\} \\ &= \arg\min_{\mathbf{y}} \left\{ \|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{y} - \mathbf{g}\|_2^2 \right\} \end{aligned} \tag{2.55}$$

Setting the derivative with respect to $\mathbf{y}$ zero, one obtains

$$\text{prox}_f(\mathbf{x}) = \frac{\mathbf{x} + \lambda\mathbf{g}}{1 + \lambda} \tag{2.56}$$

Example $f(\mathbf{x}) = \lambda \cdot \|\mathbf{x} - \mathbf{g}\|_1$

This is the L_1 data term version used in the TV-L_1 model, used in the example of image denoising (Section 2.3.2). Derivation of this term based on case differentiation

of $\mathbf{x} - \mathbf{g}$ $[>, <, =]$ 0 yields the so called *soft-thresholding operator* or *shrinkage operator*

$$\text{prox}_f(\mathbf{x}) = \begin{cases} \mathbf{x} - \lambda & \text{if } \mathbf{x} - \mathbf{g} > \lambda \\ \mathbf{x} + \lambda & \text{if } \mathbf{x} - \mathbf{g} < -\lambda \\ \mathbf{g} & \text{if } |\mathbf{x} - \mathbf{g}| < \lambda \end{cases} \tag{2.57}$$

Implementation details

In our case of minimizing the Total Variation across a 2D depth map, the linear operator K expresses the first order derivatives in x and y dimension across the image

$$K = \begin{pmatrix} \nabla_x \\ \nabla_y \end{pmatrix} \tag{2.58}$$

The depth map (or image) to solve for is represented as vector of size $\mathbf{u} \in \mathbb{R}^{MN}$ with M the number of image rows and N the number of image columns. The Total Variation operator in Equation 2.58 is then formally a sparse 2-dimensional matrix of size $\mathbb{R}^{2MN \times MN}$ with

$$\nabla_x = \begin{pmatrix} -1 & 1 & 0 & & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & \\ & & 0 & -1 & 1 & 0 \\ & & & 0 & -1 & 1 \\ 0 & \dots & & 0 & 0 & 0 \end{pmatrix} \tag{2.59}$$

and

$$\nabla_y = \begin{pmatrix} -1 & 0 & \dots & 0 & 1 & 0 & & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & \ddots & & & & \\ 0 & \dots & & 0 & & & & \dots & 0 \end{pmatrix} \tag{2.60}$$

In the analytical derivation of the primal-dual scheme above, we require the gradient ∇ and divergence operator $\nabla^T = -\text{div}$ to be negative adjoint, such that $\langle \nabla \mathbf{u}, \mathbf{p} \rangle = - \langle \mathbf{u}, \text{div } \mathbf{p} \rangle$. Therefore we use finite forward differences with Neumann boundary conditions for the gradient operators and for the divergence operators finite backward difference with Dirichlet boundary conditions:

2D forward differences (with Neumann boundary conditions)

$$
\begin{aligned}
(\nabla_x \mathbf{u})_{ij} &= \begin{cases} \mathbf{u}_{i,j+1} - \mathbf{u}_{i,j} & \text{if } j < N-1 \\ 0 & \text{if } j = N-1 \end{cases} \\
(\nabla_y \mathbf{u})_{ij} &= \begin{cases} \mathbf{u}_{i+1,j} - \mathbf{u}_{i,j} & \text{if } i < M-1 \\ 0 & \text{if } i = M-1 \end{cases}
\end{aligned} \tag{2.61}
$$

2D backward differences (with Dirichlet boundary conditions)

$$
\begin{aligned}
(\mathrm{div}_2 \mathbf{p})_{i,j} = &\begin{cases} p^1_{i,j} - p^1_{i,j-1} & \text{if } 0 < j < N-1 \\ p^1_{i,j} & \text{if } j = 0 \\ -p^1_{i,j-1} & \text{if } j = N-1 \end{cases} \\
&+ \begin{cases} p^2_{i,j} - p^2_{i-1,j} & \text{if } 0 < i < M-1 \\ p^2_{i,j} & \text{if } i = 0 \\ -p^2_{i-1,j} & \text{if } i = M-1 \end{cases}
\end{aligned} \tag{2.62}
$$

Chapter 3

ADMM

Summary

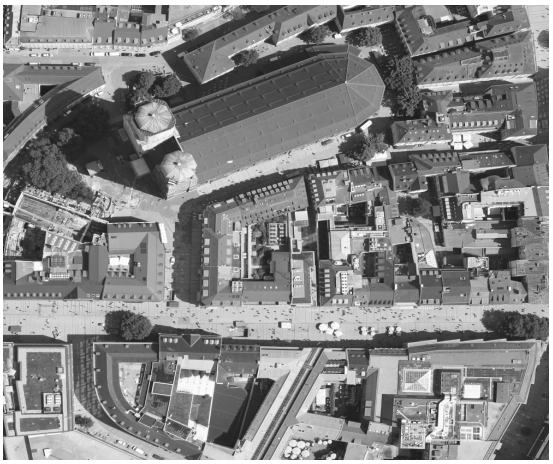
This work presents a fast algorithm for high-accuracy large-scale outdoor dense stereo reconstruction of man-made environments. To this end, we propose a structure-adaptive second-order Total Generalized Variation (TGV) regularization which facilitates the emergence of planar structures by enhancing the discontinuities along building facades. As data term we use cost functions which are robust to illumination changes arising in real world scenarios. Instead of solving the arising optimization problem by a coarse-to-fine approach, we propose a quadratic relaxation approach which is solved by an augmented Lagrangian method. This technique allows for capturing large displacements and fine structures simultaneously. Experiments show that the proposed augmented Lagrangian formulation leads to a speedup by about a factor of 2. The brightness-adaptive second-order regularization produces sub-disparity accurate and piecewise planar solutions, favoring not only fronto-parallel, but also slanted planes aligned with brightness edges in the resulting disparity maps. The algorithm is evaluated and shown to produce consistently good results for various data sets. [1]

Contributions
The main author did all the following theoretical, implementation and evaluation work on his own, plus additional conceptual discussions with the co-authors:

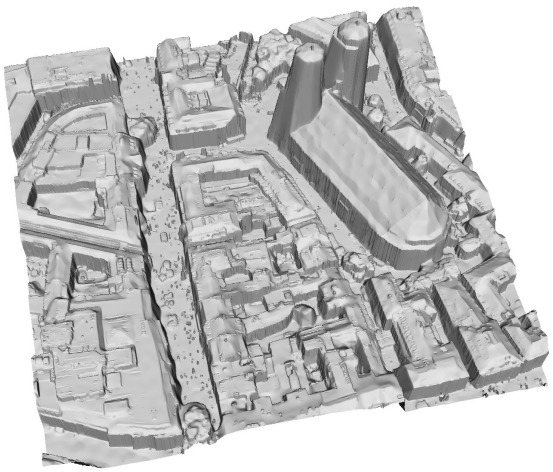
- Solving second-order TGV together with non-convex data-terms into a non coarse-to-fine framework by quadratic relaxation and optimization using augmented Lagrangian
- Subdisparty accurate exhaustive search by analytical subdisparity solution
- Using adaptive regularization by edge/line image cues

[1]

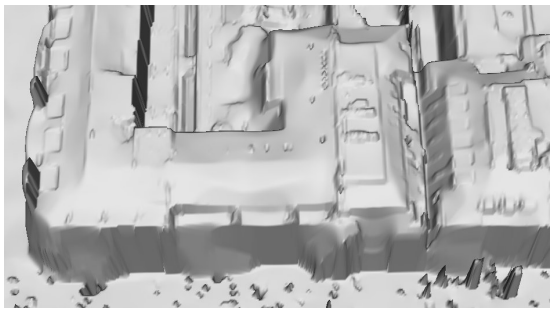
Introduction



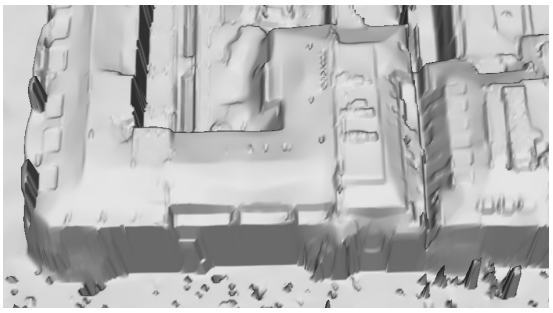
(a) Left input image



(b) Two-view 3D reconstruction



(c) Zoom-in: Reconstruction using TGV and an anisotropic diffusion tensor based on pixel-wise gradients



(d) Zoom-in: Improvements along discontinuities by additionally using high-level edge information

Figure 3.1: Detailed stereo reconstruction using two 1000×1000 wide-baseline aerial images, taking 10 seconds on common GPUs.

In the past few years, Total Variation based methods for minimizing energy functionals arising in common computer vision problems have been given a lot of attention in the research community. These algorithms are very well-suited for parallelization and, together with the recent advances of GPU-based computational power, lead to efficient algorithms, solving these optimization problems globally optimal. Recently published work solving e.g. the optical flow or stereo estimation problem can be found in [120], [107], [86], [89]. Total Generalized Variation (TGV) was originally introduced in [14] as a higher-order extension of Total Variation minimization (TV) and favors the solution to consist of piecewise polynomial functions

(e.g. fronto-parallel, affine, quadratic). Like the original TV formulation, the TGV regularizer also is convex and allows for computation of the global optimum. In the following two years, the second-order variant of TGV has been applied to depth map fusion in [87] and dense stereo estimation in [89], basically assuming that the surface to reconstruct is locally planar and not implying fronto-parallel constraints only. For being able to use robust cost functions which are usually highly non-linear, a typical choice is to linearize the costs inside a coarse-to-fine strategy (see e.g. [89]). The main drawback of this approach is that fine scene-details which are not captured in the lower pyramid levels are highly likely to be missing completely in the final reconstruction. Applying TGV as regularizer for stereo estimation, the energy functional we will use throughout the rest of the paper and need to minimize reads

$$E = \int_{\Omega} \{\lambda_s |G(\nabla \mathbf{u} - \mathbf{v})| + \lambda_a |\nabla \mathbf{v}| + \lambda_d C(\mathbf{u})\} \, \mathrm{d}\mathbf{x} \tag{3.1}$$

with $\mathbf{u}(\mathbf{x}) \in \Gamma$ the disparity/depth map to solve for (Γ being the disparity search space), an additional vector field $\mathbf{v}$ and Ω being the image space $\mathbb{R}^{M \times N}$. Note that for brevity, we just write $\mathbf{u}, \mathbf{v}$ instead of $\mathbf{u}(\mathbf{x}), \mathbf{v}(\mathbf{x})$. So instead of just enforcing the norm of the gradient of $\mathbf{u}$ to be minimal, which equals favoring fronto-parallel surfaces, the additional vector field $\mathbf{v}$ gets subtracted from the gradient of $\mathbf{u}$ and in turn is also forced to have low variation. Therefore, piecewise affine functions are being favored, as these have a constant gradient whose derivative tends to zero. The values $\lambda_s, \lambda_a, \lambda_d$ are scalar weights and balance the impact of the smoothness term, the affine term and the data term.

The linear operator G in Equation 3.1 serves to adapt the amount of regularization locally, depending on some information derived from the input images. A famous choice for G is for example the anisotropic Nagel-Enkelmann operator [74], which, in addition to the original paper, has been widely used and modified throughout the literature ([120], [89]). However, all these methods have in common, that they compute an adaptive regularization weight based on the local image gradient at the considered pixel solely. This usually improves the sharpness along discontinuities, but does not necessarily impose straight edges along man made structures. To improve the accuracy of the stereo estimation along these straight-line discontinuities, we integrate an adaptive regularization weight based on detected high-level line segments, which is inherently easy to integrate into the proposed global optimization framework.

Unfortunately, we cannot solve Equation 3.1 directly with e.g. a primal-dual gradient based approach [86], since the data term should be a strong and reliable cost function to fit our needs of being robust against some amount of change in perspective and illumination (and therefore in general non-convex). This problem often is bypassed by linearizing the cost function and solving the resulting convex problem. Since this 1st order Taylor approximation of the cost function is only valid locally, the whole algorithm needs to be wrapped into a coarse-to-fine warping framework

[15], which we explicitly want to avoid to not loose fine structures already in the coarsest level. In the following section, we will explain our solution to this minimization problem.

Edge-segment based adaptive regularization

The anisotropic diffusion tensor G in Equation 3.1 serves the purpose of an anisotropic weighting of the regularizer based on the image gradient. It enforces low regularization/smoothness along image edges, and high smoothness in homogenous image regions. It is based on the Nagel-Enkelmann operator [74] and was proposed in [120]:

$$G = \exp(-a \cdot |\nabla I_{ref}|^b) \cdot nn^T \; + \; n^{\perp} n^{\perp T} \tag{3.2}$$

with the direction of the image gradient $n = \binom{n_x}{n_y} = \frac{\nabla I_{ref}}{|\nabla I_{ref}|}$, an perpendicular vector $n^{\perp}$ and weighting parameters a, b.
However, as this diffusion tensor is based on pixelwise gradients (incorporating spatial context to a minor degree by a prior Gaussian convolution), it does not provide a strong and consistent regularization direction for small low-contrast edges as shown in Figure 3.2.
Using high-level edge segments as additional a priori information is a logical choice for guiding the optimization framework to straight-line discontinuity reconstructions. However, the main problem with this approach is the robustness of the edge detection, as for most edge detection algorithms (e.g. Canny [18]), textured regions result in a high edge density and therefore many false detections. A second problem for heterogenous image data is the need to manually tune the parameters for each group of images separately, to obtain reasonable results.
The recently introduced Fast Line Segment Detector (LSD) [117] adresses both of these problems and gives outstanding results while being computationally quite efficient. The integration of the edge-segments into the optimization framework is straight forward, as we repeat the process described in Equation 3.2 with the Gauss-convoluted binary mask of detected edge segments as input image, resulting in a second diffusion tensor G'. We obtain the combined diffusion tensor by updating the values of G with the values of G' at the position of detected lines (see Figure 3.2).

Fast optimization by quadratic splitting and augmented Lagrangian

In [107], a quadratic relaxation between the convex regularizer and the non-convex data term was proposed for minimizing a Total Variation based optical flow energy functional and [75] used a similar approach for image driven and TV-based stereo estimation. We build upon these ideas and split the image driven TGV stereo problem from Equation 3.1 into two subproblems and, using quadratic relaxation, couple the convex regularizer $R(\mathbf{u})$ and non-convex data term $C(\mathbf{u})$ through an auxiliary variable $\mathbf{a}$:

$$E = \int_{\Omega} R(\mathbf{u}) + C(\mathbf{a}) + \frac{1}{2\theta}(\mathbf{u} - \mathbf{a})^2 \, \mathrm{d}\mathbf{x} \ . \tag{3.3}$$

By iteratively decreasing $\theta \to 0$, the two variables $\mathbf{u}, \mathbf{a}$ are drawn together, enforcing the equality constraint $\mathbf{u} = \mathbf{a}$.
As an alternative, we incorporate this equality constraint not uniformly for each pixel, but via an additional augmented Lagrange multiplier $\mathbf{L}$ (see e.g. [5]) and optimize for it as well. The resulting energy minimization problem based on Equation 3.3 then reads as follows

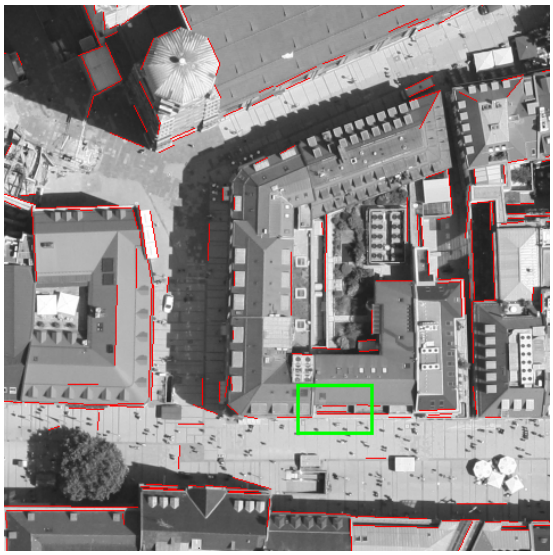
$$\begin{aligned} \mathbf{u} = \arg\min_{\mathbf{u}} \ \{ \ & \lambda_s |G(\nabla \mathbf{u} - \mathbf{v})| + \lambda_a |\nabla \mathbf{v}| + \lambda_d C(\mathbf{a}) + \\ & \mathbf{L}(\mathbf{u} - \mathbf{a}) + \frac{1}{2\theta}(\mathbf{u} - \mathbf{a})^2 \ \} \end{aligned} \tag{3.4}$$

Our experiments showed that this improves the robustness of the algorithm w.r.t. the choice of the θ-sequence and additionally speeds up the algorithm by a factor of 2 (see Figure 3.4).
While the regularization term is convex in $\mathbf{u}$ and can be solved efficiently using a primal-dual approach for a fixed auxiliary variable $\mathbf{a}$, the non-convex data term can be solved point-wise by an exhaustive search over a set of discretely sampled disparity values. This process is done alternatingly in an iterative way.

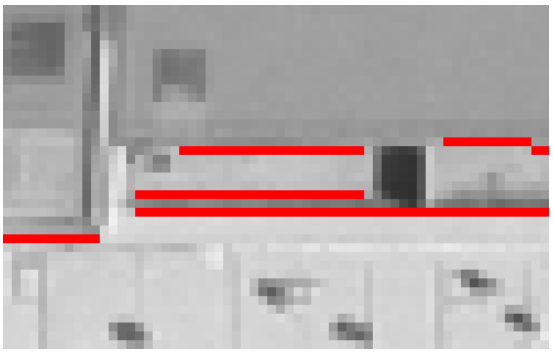
Convex solution

To solve for the disparity map $\mathbf{u} \in \mathbb{R}^{M \times N}$ (in the following written as stacked vector $\mathbb{R}^{MN \times 1}$) in the regularizer term of Equation 3.4, we need to overcome the non-differentiable L_1-norm, which complicates any gradient descent based minimization scheme. To this end we apply the Legendre-Fenchel transform to obtain the dual formulation / conjugate of our L_1 regularizers

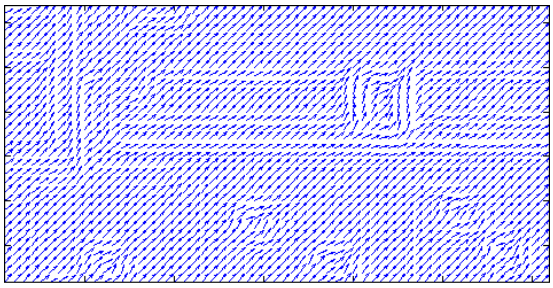
$$\lambda \|AG\mathbf{u}\|_1 = \arg\max_{\|\mathbf{p}\| \leq \lambda} \ \{\langle AG\mathbf{u}, \mathbf{p} \rangle\} \tag{3.5}$$



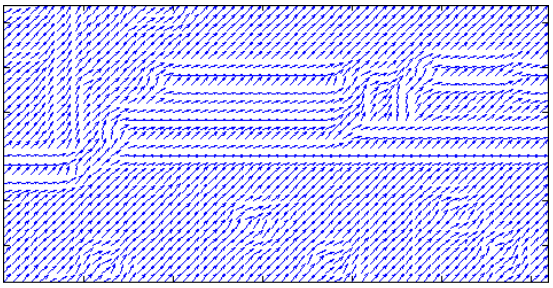
(a) Part of the left stereo image, overlayed with detected edges



(b) Zoom-in area of green rectangle



(c) Zoom-in area: Vector field of the anisotropic diffusion tensor based on pixelwise gradients



(d) Vector field additionally incorporating high-level edge information

Figure 3.2: Influence of additional high-level edge priors on the anisotropic regularization: Due to low contrast, the Nagel-Enkelmann operator in c) cannot capture the building edge of b) very well. Using additional edge information d) improves the regularization direction.

where the matrix multiplication $A\mathbf{u}$ computes the $2MN \times 1$ gradient vector and $G \in \mathbb{R}^{M \times N}$ contains the element-wise weighting factors. Applied to our problem, we obtain the conjugates

$$\begin{aligned} \lambda_s \cdot \|G(\nabla \mathbf{u} - \mathbf{v})\|_1 &= \max_{\mathbf{p} \in P} \{\langle G(\nabla \mathbf{u} - \mathbf{v}), \mathbf{p} \rangle\} \\ \lambda_a \cdot \|\nabla \mathbf{v}\|_1 &= \max_{\mathbf{q} \in Q} \{\langle \nabla \mathbf{v}, \mathbf{q} \rangle\} \end{aligned} \tag{3.6}$$

such that the saddle-point problem in the primal variables $\mathbf{u}, \mathbf{v}$ and their dual correspondences $\mathbf{p}, \mathbf{q}$ with constraints $P = \{\mathbf{p} \in \mathbb{R}^{2MN} : \|\mathbf{p}\|_\infty \leq \lambda_s\}$ and

$Q = \{\mathbf{q} \in \mathbb{R}^{4MN} : \|\mathbf{q}\|_\infty \leq \lambda_a\}$, coupled with the data term is $\max_{\mathbf{p},\mathbf{q}} \min_{\mathbf{u},\mathbf{v},\mathbf{a}}\{E\}$ with

$$E = \langle G(\nabla\mathbf{u} - \mathbf{v}), \mathbf{p}\rangle + \langle \nabla\mathbf{v}, \mathbf{q}\rangle + \lambda_d C(\mathbf{a}) + \mathbf{L}(\mathbf{u} - \mathbf{a}) + \frac{1}{2\theta}(\mathbf{u} - \mathbf{a})^2 \tag{3.7}$$

Fixing the variables $\mathbf{a}$ and $\mathbf{L}$, we obtain the minimum of Equation 3.7 for $\partial_{\mathbf{u},\mathbf{v},\mathbf{p},\mathbf{q}} E(\mathbf{u}, \mathbf{v}, \mathbf{a}, \mathbf{p}, \mathbf{q}) = 0$ and using an iterative gradient descent in the primal variables and gradient ascent in the dual variables.

Non-convex solution

To solve for the auxiliary variable $\mathbf{a}$ in the data term of Equation 3.4, we keep the variables $\mathbf{u}, \mathbf{L}$ fixed and perform a point-wise exhaustive search over all $\mathbf{a}(\mathbf{x}) \in \Gamma$

$$\min_{\mathbf{a}(\mathbf{x})\in\Gamma} \left\{ \lambda_d C(\mathbf{a}) + \mathbf{L}(\mathbf{u} - \mathbf{a}) + \frac{1}{2\theta}(\mathbf{u} - \mathbf{a})^2 \right\} \tag{3.8}$$

Note that in order to retain the TGV smoothness, it is necessary to perform the exhaustive search using subdisparity sampling steps. As this may look computational expensive at first glance, it does not affect the overall performance in a measurable way if implemented with care (see Section 3.5).

Augmented Lagrangian update

According to e.g. [5], the Lagrange multiplier L is updated by $\mathbf{L}^{n+1} = \mathbf{L}^n + \frac{1}{2\theta^n}(\mathbf{u} - \mathbf{a})$, with the augmented penalty function $\frac{1}{2\theta^n}$ monotonically increasing as $\theta^n \to 0$.

Algorithm

In this section we will describe how to solve the energy minimization problem stated in Equation 3.4. As a first step, since the stereo estimation should work with various scales in depth and different cost functions as well without having to adjust the parameters for each dataset, we initially norm both $\mathbf{u} \to [0, 1]$ and the costs $C \to [0, 1]$. Doing so, we can fix nearly all parameters internally and only need to expose the weighting factors λ_d, λ_s, balancing the impact of the data term and smoothness term, to be set by the user. After evaluating the algorithm for a variety of scenarios (indoor, ground-based outdoor, aerial) and benchmarks (see Section 5.5), we obtained the best results for $\lambda_a = 8\lambda_s$ and fix this value to not bother the user with the weighted impact of the affine term additionally.

1. Fixing $\mathbf{a}^n$ and $\mathbf{L}^n$, run the primal-dual optimization for a number of inner iterations, performing gradient ascents on the dual variables $\mathbf{p}, \mathbf{q}$ and gradient descents on the primal variables $\mathbf{u}, \mathbf{v}$:
for $i = 1 : nIterSmooth$ do

$$\begin{aligned}
\mathbf{p}^{n+1} &= \Pi_P\left(\mathbf{p}^n + \tau_p G(\nabla\hat{\mathbf{u}}^n - \hat{\mathbf{v}}^n)\right)\\
\mathbf{q}^{n+1} &= \Pi_Q\left(\mathbf{q}^n + \tau_q \nabla\hat{\mathbf{v}}^n\right)\\
\mathbf{u}^{n+1} &= \Pi_U\left(\frac{\mathbf{u}^n + \tau_u \mathrm{div}\left(G\mathbf{p}^{n+1}\right) - \tau_u\mathbf{L}^n + \frac{\tau_u}{\theta^n}\mathbf{a}^n}{1 + \frac{\tau_u}{\theta^n}}\right)\\
\mathbf{v}^{n+1} &= \mathbf{v}^n + \tau_v(\mathbf{p}^{n+1} + \mathrm{div}\mathbf{q}^{n+1})\\
\hat{\mathbf{u}}^{n+1} &= 2\mathbf{u}^{n+1} - \mathbf{u}^n\\
\hat{\mathbf{v}}^{n+1} &= 2\mathbf{v}^{n+1} - \mathbf{v}^n
\end{aligned}$$

2. Fixing $\mathbf{u}^{n+1} = \tilde{\mathbf{u}}$, perform a point-wise search

$$\mathbf{a}^{n+1} = \underset{\mathbf{a}(\mathbf{x})\in\Gamma}{\arg\min}\left\{\lambda_d C(\mathbf{a}) + \mathbf{L}^n(\tilde{\mathbf{u}} - \mathbf{a}) + \frac{(\tilde{\mathbf{u}} - \mathbf{a})^2}{2\theta^n}\right\}$$

3. Update $\mathbf{L}^{n+1} = \mathbf{L}^n + \frac{1}{2\theta^n}(\mathbf{u}^{n+1} - \mathbf{a}^{n+1})$

4. If $n < n_{stop}$, update $\theta^{n+1} = \theta^n(1 - \beta n)$, $n = n + 1$, goto step (1) else stop

Algorithm 3.1: Algorithm for ADMM-based 3D reconstruction

The complete optimization of the proposed energy functional in Equation 3.4 is done iteratively as described in Algorithm 3.1, initializing the primal variable with the disparity value associated to the data cost minimum (winner-takes-all solution), $\mathbf{u}^0 = \mathbf{a}^0 = \mathrm{argmin}_{\mathbf{a}(\mathbf{x})\in\Gamma}\ C(\mathbf{x}, \mathbf{a}(\mathbf{x}))$, setting the dual variables to zero ($\mathbf{p}^0 = 0$, $\mathbf{q}^0 = 0$), and starting with iteration $n = 0$ and $\theta^0 = 1$.
To ensure that $\|\mathbf{p}\|_\infty \leq \lambda_s$ and $\|\mathbf{q}\|_\infty \leq \lambda_a$, the proximal mappings above are given as $\Pi_P(\mathbf{p}) = \frac{\mathbf{p}}{\max\{1, \|\mathbf{p}\|/\lambda_s\}}$ and $\Pi_Q(\mathbf{q}) = \frac{\mathbf{q}}{\max\{1, \|\mathbf{q}\|/\lambda_a\}}$ and for keeping $\mathbf{u}$ in valid range, we use Π_U as the truncation of $\mathbf{u}^{n+1}$ onto the interval $[0, 1]$. Also note, that in the analytical derivation of the primal-dual scheme above, we require the gradient and divergence operators to be negative adjoint, such that $\langle\nabla\mathbf{u}, \mathbf{p}\rangle = -\langle\mathbf{u}, \mathrm{div}\ \mathbf{p}\rangle$ and $\langle\nabla\mathbf{v}, \mathbf{q}\rangle = -\langle\mathbf{v}, \mathrm{div}\ \mathbf{q}\rangle$. Therefore we use finite forward differences with Neumann boundary conditions for the gradient operators and for the divergence operators finite backward difference with Dirichlet boundary conditions. The step sizes of the gradient ascents/descents are bound to the norm of the gradient/divergence operators and are set to $\tau_u = \tau_p = 1/\sqrt{12}$ and $\tau_v = \tau_q = 1/\sqrt{8}$, as detailed in [22].

The parameter β controls how fast the convex and non-convex solution are drawn together (by decreasing θ) and is fixed to $\beta = 10^{-3}$, while the whole algorithm stops, if $n > 80$. For the number of primal-dual iterations, we set $nIterSmooth = 150$.

As already mentioned in Section 3.4, retaining the subdisparity smoothness resulting from the continuous TGV solution requires subdisparity accurate results of the exhaustive search as well. Therefore, after obtaining an integer solution for the disparity $\mathbf{a}$ which minimizes the energy

$$\arg\min_{\mathbf{a}} \left\{ \lambda_d C(\mathbf{a}) \; + \; \mathbf{L}(\mathbf{u}-\mathbf{a}) \; + \; \frac{1}{2\theta}(\mathbf{u}-\mathbf{a})^2 \right\} \quad , \tag{3.9}$$

we compute the subdisparity solution as the minimum of a parabola, fitted through the obtained integer minimum and its adjacent values at ± 1 disparities (see Figure 3.3). Parametrizing the parabola as $C(\mathbf{a}+t) = at^2 + bt + c$, the coefficients are computed using the abovementioned 3 datapoints and corresponding $t \in \{-1, 0, 1\}$. Substituting $\tilde{\mathbf{a}} = \mathbf{a} + t$, $C(\tilde{\mathbf{a}}) = at^2 + bt + c$ and optimizing for the parameter t, we obtain the subdisparity refinement $\tilde{t} \in [-\frac{1}{m}, \; \frac{1}{m}]$ as

$$\tilde{t} = \frac{\frac{\mathbf{u}-\mathbf{a}}{\theta m} - \lambda b - \frac{\mathbf{L}}{m}}{\left(2\lambda a + \frac{1}{\theta m^2}\right)} \quad , \tag{3.10}$$

with $m = |\Gamma|$ being the number of disparities.

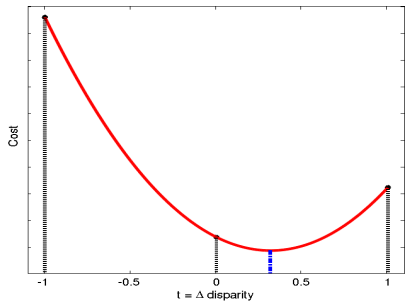


Figure 3.3: Subdisparity accurate results are required in the exhaustive search step, to retain the continuous solution of the prior TGV step

Finally, due to its iterative and locally confined computations per iteration, the algorithm is very well-suited for parallelization and therefore implemented on GPU.

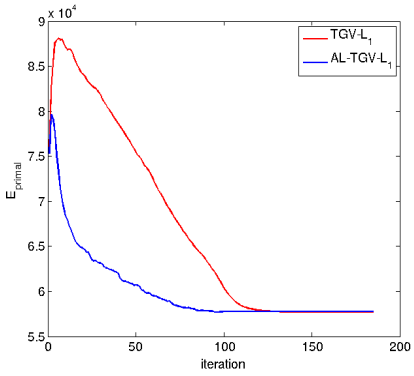


Figure 3.4: Evolution of the primal energy of Equation 3.3, with and without augmented Lagrangian. The runtime is dominated by the primal-dual algorithm, such that the additional Lagrange multiplier $\mathbf{L}$ has a neglectable influence and the runtime per iteration is basically the same for the two algorithms.

Evaluation

Our algorithm is evaluated on three different data sets and in case RGB images are available, only the gray image will be used. If more than two views are available, only two of them will be used, in order to demonstrate our algorithm on two-view stereo scenarios. For all datasets, we used the Census transform [125] with windows size 7×7 as cost function, since it is quite robust to a wide range of illumination changes. Additionally, we locally aggregate the costs using Adaptive support-weights [124] with radius 7 to reduce the effect of foreground fattening, but keeping the radius quite small so as not to put too much fronto-parallel assumption into the cost window. For regularization we are using two parameter sets: $\{\lambda_d = 1.0,\ \lambda_s = 0.2\}$ for the low resolution Middlebury stereo benchmark [99] and $\{\lambda_d = 0.4,\ \lambda_s = 1.0\}$ for the KITTI stereo benchmark [34] and the aerial images. The algorithm was run on a Nvidia GTX 680 GPU to which all given runtime performances relate to.

Middlebury benchmark: The Middlebury stereo benchmark [99] provides an additional discontinuity mask which we will use for the evaluation of our edge-segment based adaptive regularization. In Table 3.1 and Figure 3.5 we show the results of our algorithm both with the adaptive edge-segment regularization switched on and without. For all scenes except the teddy data set the results improve along the discontinuity regions, whereas for the teddy dataset results are worsening on the strongly slanted plane at the very bottom of the image. We are using the same parameters and cost functions described in Section 5.5 for all data sets and only take the gray value images of the stereo pairs as input.

Algorithm	Tsukuba			Venus			Teddy			Cones		
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
TGV	3.66	4.33	12.0	0.21	1.00	2.88	3.93	9.66	12.1	2.44	11.1	7.20
TGV+edge	3.58	4.21	11.6	0.19	1.01	2.61	4.30	9.95	13.0	2.41	11.2	7.01

Table 3.1: Results of the proposed algorithm for the Middlebury Stereo benchmark (bad pixel ratio for errors $>$ 1px), once without an anisotropic diffusion tensor (TGV), once with the combined diffusion tensor of Section 3.3 (TGV+edge).

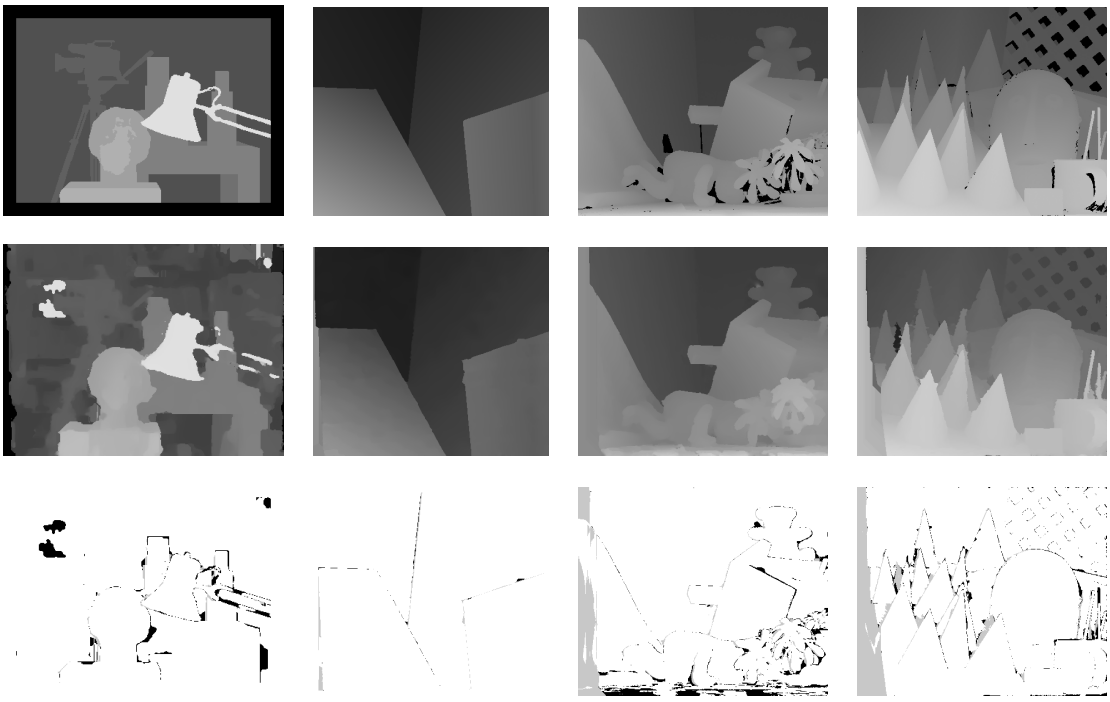


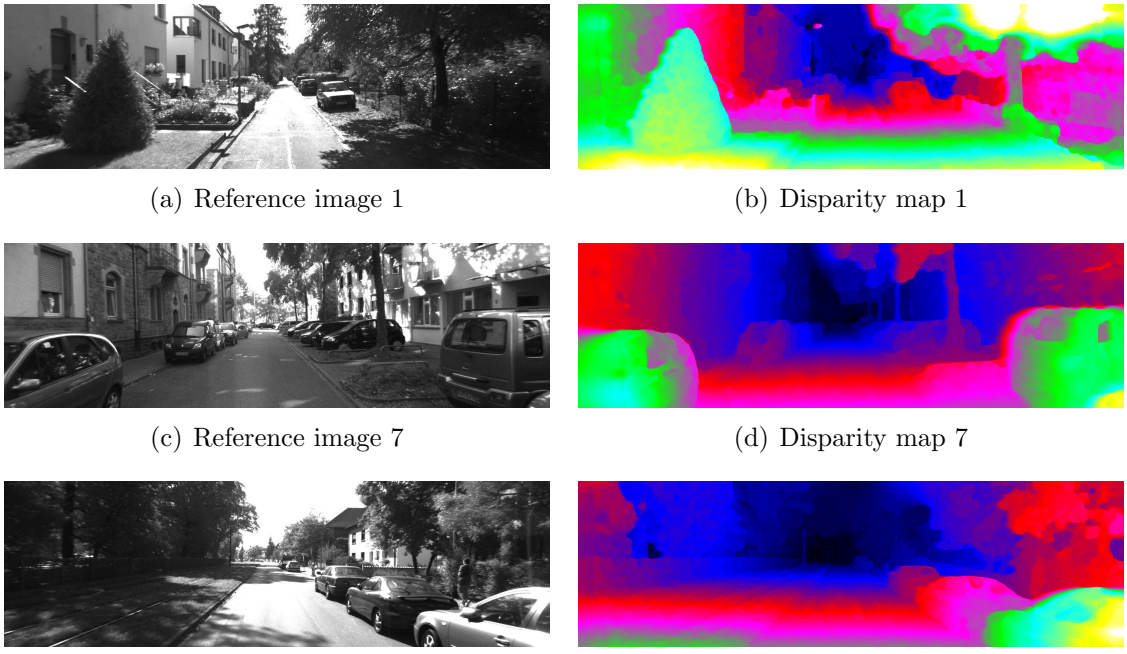
Figure 3.5: Results of the proposed algorithm for the Middlebury Stereo benchmark. Top row: ground truth, Middle row: our results, bottom row: bad pixel areas in black (threshold = 1px). The parameters are identical for all data sets and only the gray value images were taken.

KITTI benchmark: In contrast to the 4 test images of the Middlebury benchmark above, where the disparity search range is very small, the environment highly textured and the illumination conditions nearly constant, the KITTI stereo Benchmark [34] consists of 195 very challenging stereo images from ground based outdoor scenarios, together with ground truth obtained by laser scanning. In total, we achieve rank 11 in the benchmark, with a runtime of 20s per image. Additionally, we compare our results against the closest related published algorithms, also based on minimizing higher-order Total Variation (see Table 3.2). While we outperform

the coarse-to-fine based ITGV algorithm [89] in terms of accuracy, we do not yet quite achieve the accuracy of the functional lifting based ATGV algorithm [90]. For some exemplary results of the proposed algorithm see Figure 3.6.

Rank	Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime
8	ATGV	**5.05%**	6.91%	**1.0 px**	1.6 px	6 min
11	Proposed	5.48%	**6.60%**	1.1px	**1.2px**	20s
17	ITGV	6.31%	7.40%	1.3px	1.5px	**7s**

Table 3.2: Results for the challenging KITTI stereo benchmark [34] (195 outdoor stereo pairs). The bad pixel ratio of *Out-Noc, Out-All* is the common 3px threshold. For comparison, we further added the closest related algorithms as well. For some exemplary results see Figure 3.6.



(a) Reference image 1 (b) Disparity map 1

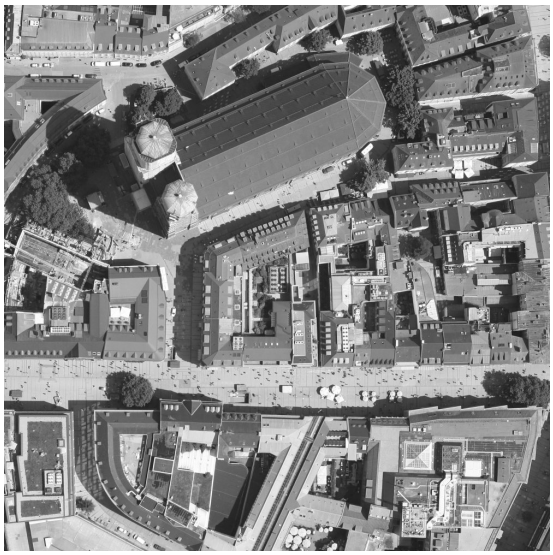
(c) Reference image 7 (d) Disparity map 7

(e) Reference image 15 (f) Disparity map 15

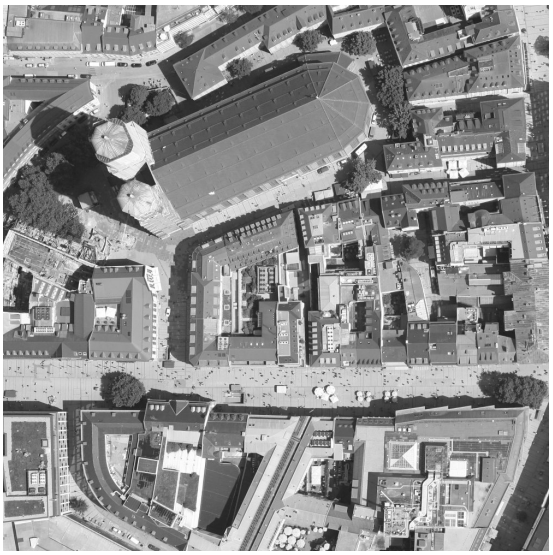
Figure 3.6: Example results for the KITTI stereo benchmark. From top to bottom: bad, medium and good results

Aerial imagery: In a third data set, we apply our algorithm to aerial imagery. Despite usually having numerous overlapping images, covering every point of the

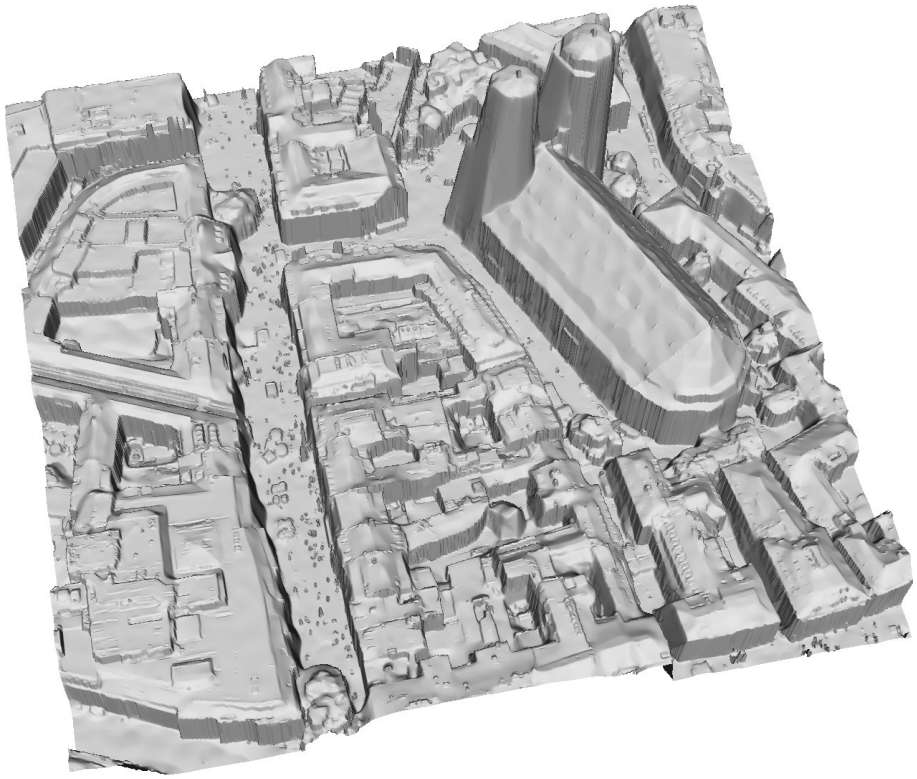
scene manifold, we concentrate on showing the potential of the proposed algorithm on single stereo pairs, and apply no fusion of the resulting heightmaps in this paper. In contrast to the rectified images given in the abovementioned benchmarks, in this data set we have camera models ready for each input image, allowing us to evaluate the cost function at constant intervals in object space (using a plane-sweep approach) instead of sampling at constant disparity intervals. Thus our algorithm can treat every height value equally, whereas in disparity space, small changes in low disparities result in bigger height-differences than changes in large disparities. In Figure 3.7, the resulting 3D reconstruction is shown together with the two stereo images. The proposed method clearly preserves very fine structural details of the 3D scene (e.g. roof structures), while at the same time smoothing locally planar surfaces (church roof) quite well.



(a) Left image



(b) Right image



(c) Stereo reconstruction

Figure 3.7: a), b) Two wide-baseline aerial images (≈15cm ground resolution) c) Resulting heightmap (in camera coordinate system, not in orthogonal UTM coordinate system) of two-view stereo estimation using the proposed algorithm. Please note the fine roof structures in the 3D reconstruction, but the outliers due to moving people as well. The computation time for a 1000×1000 image using 100 disparity values is about 10s (using a Nvidia GTX 680 GPU).

Conclusion

In this paper we proposed an algorithm for large-scale high-accuracy stereo reconstruction of man-made worlds. To this end, we combine a non-convex data term which is robust to real-world illumination changes with a regularizer which exploits the fact that man-made worlds (buildings, cities, etc.) exhibit a large number of planar facades. The regularizer is an adaptive second-order total generalized variation modulated by means of an edge-indicator. We propose an optimization scheme consisting of a quadratic decoupling combined with an augmented Lagrangian approach which alternatingly solves the problems of correspondence finding and structure-adaptive regularization. Experiments show that the proposed augmented Lagrangian approach is faster by about a factor of 2. Validations on established stereo benchmarks and large-scale aerial images show that the proposed method provides substantial improvements over the standard TGV regularization leading to highly-accurate reconstruction of large-scale scenes.

Chapter 4

Dense SLAM

Summary

In this work we propose an algorithm for dense and direct large-scale visual SLAM that runs in real-time on a notebook. A variational dense 3D reconstruction algorithm was developed which robustly integrates data terms from multiple images. This mitigates the effect of the aperture problem and is demonstrated on synthetic and real data. An additional property of the variational reconstruction framework is the ability to integrate sparse depth priors into the early stages of the visual depth reconstruction, leading to an implicit sensor fusion scheme for a variable number of heterogenous depth sensors. Embedded into a keyframe-based SLAM framework, this results in a memory efficient representation of the scene and enables us to densely reconstruct large scenes in real-time. Experimental validation on the KITTI dataset shows that our method can recover large-scale and dense reconstructions of entire street scenes in real-time from a driving car. [1]

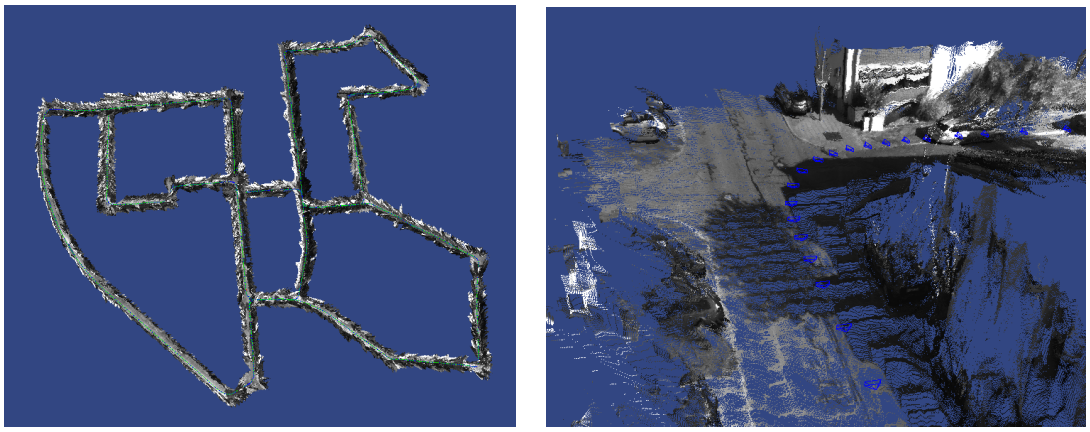
Contributions
The main author did all the following theoretical and implementation work regarding the depth reconstruction on his own, whilst the integration into the SLAM framework and its corresponding evaluation was done together with Aljaz Bozic, plus conceptual discussions with the co-authors:

- Realtime implementation of dense variational 3D reconstruction
- Strengthening the data term by adding information from multiple input images into the stereo reconstruction optimization framework and solving for them in a robust way by minimizing the sum of their L1-norms
- Adding optional depth priors into the stereo reconstruction framework

[1]

Introduction and Related Work

The reconstruction of the world from moving cameras has seen an enormous progress over the last few years: Algorithms have become increasingly robust, fast and large-scale capable. While active sensors such as lasers or RGB-D cameras have become a popular means to obtain dense reconstructions of the world, in this work we focus on the case of color cameras as these are more prevalent and cheaper than lasers and not limited to indoor and near-range settings as current RGB-D cameras. Due to an increasing (and redundant) amount of sensors built into nowadays moving camera systems (e.g. cars and smartphones), we want to design our optical stereo reconstruction in such a way that integration of sparse depth priors arising from different depth sensors can be used to guide the stereo reconstruction in areas where cameras usually perform bad (oversaturated and/or textureless areas). To increase robustness against wrong matches, the reconstruction process further should be inherently able to use multiple images (if available) instead of just two.



(a) Final map (b) Detailed zoom

Figure 4.1: Dense large-scale reconstruction for an automotive image sequence.

Related Work

The visual simultaneous localization and mapping, often called structure-and-motion, is traditionally solved by extracting and tracking a set of keypoints in order to facilitate real-time performance. A premier example of real-time large-scale dense reconstructions obtained on the basis of a keypoint-based visual-inertial SLAM system was recently proposed in [102].

With recent improvements in computing hardware and algorithms, so-called direct methods for visual SLAM have been promoted. Rather than precomputing an

invariably heuristic sparse subset of keypoints, they directly rely on all available image information in order to recover structure and motion. Such direct Visual SLAM methods include [110, 75, 36, 31, 28, 81, 52].

Another important aspect for large-scale dense reconstruction is the representation of the dense 3D geometry itself. Surfel-based methods [109, 121] are a natural extension of pointclouds by extending each point to a typically ellipsoidal shape. Spatial regularization of the 3D model however is non-trivial as with normal pointclouds. While volumetric representations based on regular voxel grids [36, 52, 43] typically yield the highest accuracy due to easy data fusion and spatial regularization, they suffer from severe memory limitations. These limitations can be overcome by using octrees [106, 115] or voxel hashing [76, 102]. Yet, these representations still – as with the regular voxel grid – cannot easily handle loop-closures and the entailing pose refinements. Keyframe-based approaches [70, 28] are naturally able to optimize the camera poses when correcting for loop-closures, depth information from nearby depthmaps can be fused into them and they provide a regular grid to facilitate spatial regularization.

The most closely related works fall into two categories: On one hand, there are direct visual SLAM techniques which compute dense geometry in real-time [110, 75], albeit not at a large scale – typically only smaller desktop environments are recovered because for larger-scale structures substantial drift in the frame-to-frame tracking tends to create distortions. On the other hand, there are visual SLAM algorithms like LSD-SLAM [28] which do compensate for drift by pose graph optimization. While they can recover environments of many buildings and street scenes in real-time, the corresponding reconstructions are not dense. As a consequence, they may not be ideal for obstacle avoidance, path planning and a complete visualization of the environment. In this work we combine the strengths of both of these approaches and propose a direct visual SLAM system which recovers large-scale dense environments in real-time on commodity notebook hardware.

As for the dense stereo reconstruction – in this work we focus on a variational formulation, as this approach provides a general and modular framework with additional prior knowledge/assumptions easy to integrate into the overall energy minimization. This prior knowledge is not restricted to smoothness assumptions like fronto-parallel surfaces, or slanted plane [14], [35], but also allows for pixelwise data-priors n [87].
Variational methods for semi-real-time dense stereo have been already been developed by e.g. [110], [75], [81], [35]. However, [110] achieve semi-real-time framerates of 20 fps only when performing the corresponding optimization on the GPU inexactly in shared memory and thereby creating small blocking artifacts, [75] computing a full cost volume (disparity space image) over the complete disparity search range (which must be known and fixed in advance), [81] just running a variational denoising on the computed noisy depth map and [35] achieving semi-real-time capability

of 10fps for 640×480 images. All of the methods above treat the computation of a single depth map individually and do not take the temporal correlation between successive depth maps into account, when handling a video stream of successive frame. Despite their convex formulation, they all start the optimization from a blank initial guess, thereby needing more time for convergence as if an approximate solution would already be available.

Contributions and Overview

To achieve the abovementioned goals we propose the following key contributions:

- Our main contribution is the addition of a real-time capable dense 3D reconstruction into a SLAM framework which is based on direct image alignment for camera tracking and loop-closure correction by pose-graph optimization on the keyframe representation of the 3D scene. The dense reconstruction is based on a variational approach, imposing structural smoothness priors onto the scene geometry and capable of inpainting textureless areas where the corresponding image matching provides no meaningful data term.

- To strengthen the data term in non-discriminative cases, we propose to directly include the image matching information of multiple images into the reconstruction step by minimizing the sum of their absolute values. The advantage of minimizing the sum of L1 terms is in effect a median-based solution compared to the mean-based solution when just computing the summed average. This results in improved robustness against erroneous matches.

- Formulating the 3D reconstruction as a variational energy functional implicitly enables us to add additional prior depth information into our model, originating e.g. in sparse 3D feature reconstruction or laser scanner. This provides an elegant sensor fusion model for heterogenous depth sensors.

The tracking of the camera positions is performed by direct image alignment, with an additional check for loop-closures resulting in a pose-graph optimization using on the constraints between keyframe poses. Tracking and pose refinement are done by using the publicly available implementation of [28] and we refer the readers for further details to their work.

Dense Depth Reconstruction

Multi-View Data Terms and Sparse Priors

We model our 3D reconstruction of the depthmap $\mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ as the minimization of the energy functional

$$E = \int_{\Omega} R(\mathbf{u}) + \sum_{i}^{K} C_i^{(d)}(\mathbf{u}) + \sum_{i}^{S} D(\mathbf{u}, \mathbf{v}_i) \tag{4.1}$$

where $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ denotes the image space, $C_i^{(d)}$ are the data terms of image matching costs for multiple images $I_{1..K} : \Omega \rightarrow \mathbb{R}$, D is the cost function for forcing the solution $\mathbf{u}$ to be similar to various sources of prior depth information $\mathbf{v}_{1..S}$, and $R(\mathbf{u})$ is a regularizer enforcing spatial smoothness of the solution. When discretizing the energy for numerical optimization

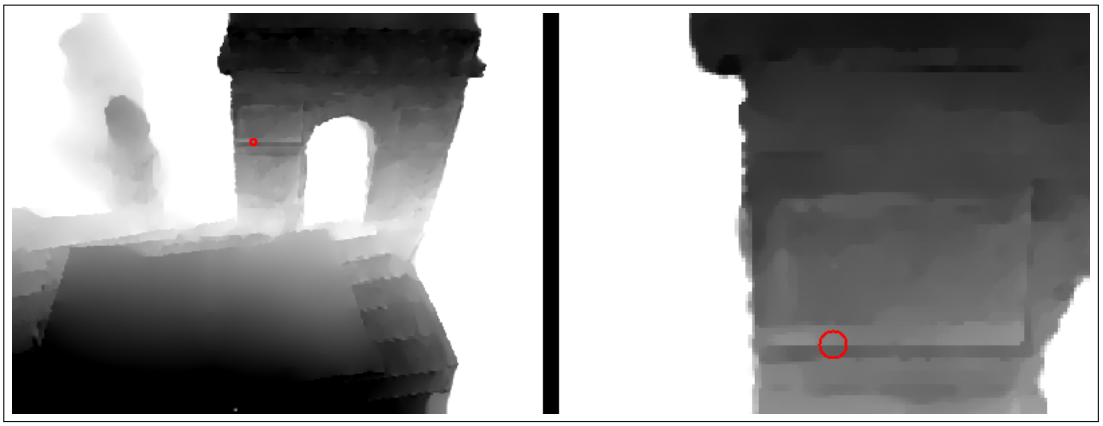
$$E = |\nabla \mathbf{u}|_{\epsilon} + \sum_{i=1}^{K} \lambda_i^{(d)} \left|\mathbf{C}_i(\mathbf{u})\right|_1 + \sum_{i=1}^{S} \lambda_i^{(p)} \left|\mathbf{u} - \mathbf{v}_i\right|_1 \tag{4.2}$$

we employ the L_1-norm to both the K image matching data terms and the S a priori data terms for being able to robustly cope with outliers. As regularizer we choose to favor locally smooth structures and minimize the Total Variation, with the Huber norm $|.|_{\epsilon}$ both preserving discontinuities at sharp object transitions and locally avoid staircasing artifacts. When having detailed estimates about the image matching confidence or the a priori data terms $\mathbf{v}_i$, we can directly integrate this knowledge into the corresponding weighting factors $\lambda_i^{(d)}$ and $\lambda_i^{(d)}$.

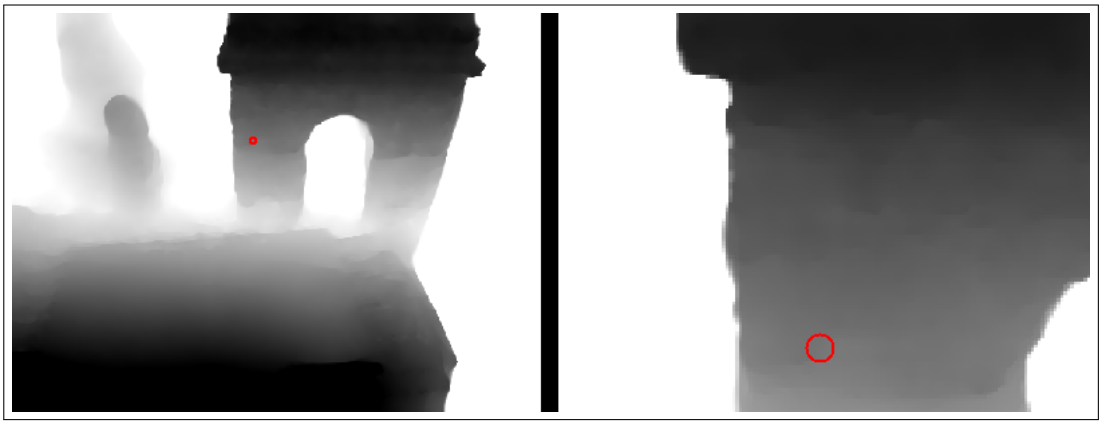
The data fidelity functions $\mathbf{C}_i$ measure how well the reference image $I_{ref}(\mathbf{x})$ matches the warped images $I_i(\omega(\mathbf{x}, \mathbf{u}(\mathbf{x})))$ and are given as simple intensity differences

$$\mathbf{C}_i(\mathbf{x}, \mathbf{u}(\mathbf{x})) = I_i(\omega(\mathbf{x}, \mathbf{u}(\mathbf{x}))) - I_{ref}(\mathbf{x}) \tag{4.3}$$

as this allows us to analytically computing the derivatives of the cost functions later on, instead of numerical differentiation adding further heuristical parameters. This brightness constancy assumption is a valid assumption for synchronized stereo images and for monocular images with a small temporal interframe distance when e.g. taken at typical 30 fps. For matching images over wider baselines more illumination invariant cost functions with a spatial support radius need to be chosen (e.g. the Census transform [125] or its modified scale-robust version [88]). Minimizing over multiple image matching cost functions is of special importance w.r.t. the aperture problem, arising when the epipolar lines of the involved camera setup are parallel to the image gradient, thus resulting in a flat and non-discriminative cost function.



(a)

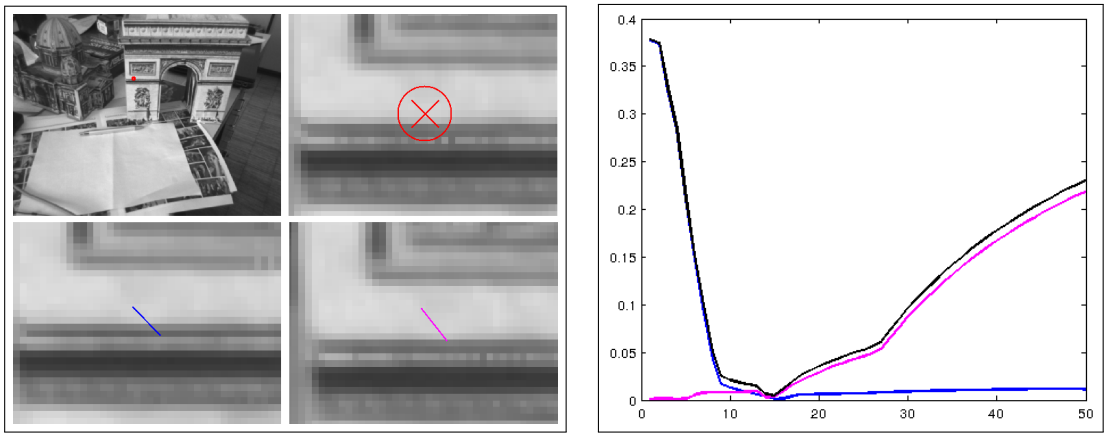


(b)

Figure 4.2: Reconstruction results using different number K of images: a) $K = 2$ b) $K = 3$. The data term was normalized to $\lambda/(K-1)$ to have equal balancing in the energy functionals.

When additional images with a different camera motion are added, the combination of different cost functions yields more information about the real depth. Especially in the typical automotive stereo camera setup, the epipolar lines of the stereo camera pair (C_t^l, C_t^r) at time step t are nearly horizontal, whereas when moving forward the epipolar lines of e.g. camera pair (C_t^l, C_{t+1}^l) are oriented rather vertically. DTAM [75] used a simple averaging of the different cost functions, resulting in one final and averaged cost function in the reconstruction process. The advantage of our L1-based solution is in effect a median-based solution compared to the mean-based solution of [75], resulting in improved robustness against erroneous matches.

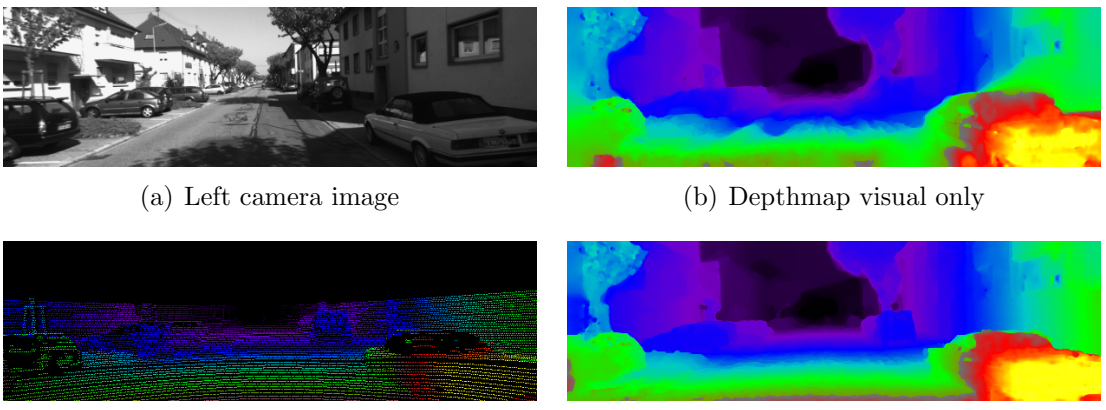
Adding sparse prior depth information $\mathbf{v}_i$ is straightforward, as we can directly



(a) (b)

Figure 4.3: Compare to Fig. 4.2: a) The corresponding pixel of interest (red) and two sampling intervals along the epipolar lines in two different images (blue and purple). As can be seen in b), both the blue and purple image matching cost function do not exhibit a clear minimum, their combination however (black) does.

add additional data constraints $\lambda_i^{(p)} |\mathbf{u} - \mathbf{v}_i|_1$ to our energy functional Eq. 4.5. Especially in an automotive setting where additional depth information from a laser scanner is given, this results in a strong prior, guiding the visual stereo reconstruction to a more accurate solution (see Fig. 4.4).



(a) Left camera image (b) Depthmap visual only

(c) Sparse laser depth data (d) Depthmap sensor fusion visual+laser

Figure 4.4: Demonstrating the influence of adding sparse laser priors early in the visual 3D reconstruction process. Only a simple coarse-to-fine warping scheme using intensity differences was used for stereo reconstruction.

Optimization

For making the minimization of the energy functional Eq. 4.5 tractable, we have to ensure its convexity, which is not yet given due to the non-convex image matching data terms $\mathbf{C}_i(\mathbf{u})$. We therefore linearize each $\mathbf{C}_i$ around an initial value $\mathbf{u}_0$ via its first order Taylor approximation

$$\mathbf{C}_i(\mathbf{u}) \approx \tilde{\mathbf{C}}_i(\mathbf{u}) = \mathbf{C}_i(\mathbf{u}_0) + (\mathbf{u} - \mathbf{u}_0)\frac{\partial \mathbf{C}_i(\mathbf{u}_0)}{\partial \mathbf{u}} \tag{4.4}$$

Because this linearization is only valid in a local range around $\mathbf{u}_0$ we need to embed the complete optimization scheme into a coarse-to-fine warping framework. The resulting energy functional

$$E(\mathbf{u}) = |\nabla \mathbf{u}|_\epsilon + \sum_{i=1}^{K} \lambda_i^{(d)} |\tilde{\mathbf{C}}_i(\mathbf{u})|_1 + \sum_{i=1}^{S} \lambda_i^{(p)} |\mathbf{u} - \mathbf{v}_i|_1 \tag{4.5}$$

is now convex, but due to the involved edge preserving and robust norms entirely non-smooth. The primal-dual algorithm of [22] provides a framework to solve certain classes of non-smooth optimization problems by transforming them to an equivalent primal-dual formulation as saddle point problem

$$\begin{aligned} &\min_{\mathbf{x}} \{F(L\mathbf{x}) + G(\mathbf{x})\} \Rightarrow \\ &\min_{\mathbf{x}} \max_{\mathbf{y}} \{\langle L\mathbf{x}, \mathbf{y}\rangle + G(\mathbf{x}) - F^*(\mathbf{y})\} \\ \Leftrightarrow &\min_{\mathbf{x}} \max_{\mathbf{y}} \left\{\left\langle L^T\mathbf{y}, \mathbf{x}\right\rangle + G(\mathbf{x}) - F^*(\mathbf{y})\right\} \end{aligned} \tag{4.6}$$

with the mapping L a linear operator, $G(.)$ and $F^*(.)$ convex functions, F^* itself the convex conjugate of F. With the goal of transforming Eq. 4.5 to the saddle point formulation of Eq. 4.6 we apply the Legendre-Fenchel transform to the norms of the

individual terms of Eq. 4.5

$$\begin{aligned} E_1(\mathbf{u}) &= \sup_{|\mathbf{p}|\leq 1} \langle \nabla \mathbf{u}, \mathbf{p} \rangle + \frac{\epsilon}{2}|\mathbf{p}|^2 \\ &= \sup_{|\mathbf{p}|\leq 1} \left\langle \nabla^T \mathbf{p}, \mathbf{u} \right\rangle + \frac{\epsilon}{2}|\mathbf{p}|^2 \end{aligned} \tag{4.7}$$

$$\begin{aligned} E_2(\mathbf{u}) &= \sup_{|\mathbf{q_i}|\leq 1} \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \tilde{\mathbf{C}}_i(\mathbf{u}), \mathbf{q}_i \right\rangle \\ &= \sup_{|\mathbf{q_i}|\leq 1} \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \left(\mathbf{C}_i(\mathbf{u}_0) + (\mathbf{u} - \mathbf{u}_0)\mathbf{C}_i'(\mathbf{u}_0) \right), \mathbf{q}_i \right\rangle \\ &= \sup_{|\mathbf{q_i}|\leq 1} \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \mathbf{u}\mathbf{C}_i', \mathbf{q}_i \right\rangle + \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \left(\mathbf{C}_i - \mathbf{u}_0\mathbf{C}_i' \right), \mathbf{q}_i \right\rangle \\ &= \sup_{|\mathbf{q_i}|\leq 1} \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \mathbf{C}_i'\mathbf{q}_i, \mathbf{u} \right\rangle + \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \hat{\mathbf{C}}_i, \mathbf{q}_i \right\rangle \end{aligned} \tag{4.8}$$

And coupling these terms together again

$$\begin{aligned} E(\mathbf{u}) = \sup_{|\mathbf{p}|,|\mathbf{q}_i|\leq 1} \Bigg\{ & \left\langle \nabla^T \mathbf{p} + \sum_{i=1}^{K} \lambda_i^{(d)} \mathbf{C}_i'\mathbf{q}_i, \mathbf{u} \right\rangle + \frac{\epsilon}{2}|\mathbf{p}|^2 + \\ & \sum_{i=1}^{K} \left\langle \lambda_i^{(d)} \hat{\mathbf{C}}_i, \mathbf{q}_i \right\rangle - \sum_{i=1}^{S} \lambda_i^{(p)} |\mathbf{u} - \mathbf{v}_i|_1 \Bigg\} \\ = \sup_{|\mathbf{p}|,|\mathbf{q}_i|\leq 1} \Bigg\{ & \left\langle L^T \cdot (\ \mathbf{p}\ \ \mathbf{q}_i\ )^T, \mathbf{u} \right\rangle + \frac{\epsilon}{2}|\mathbf{p}|^2 + \left\langle \lambda_i^{(d)} \hat{\mathbf{C}}_i, \mathbf{q}_i \right\rangle - \sum_{i=1}^{S} \lambda_i^{(p)} |\mathbf{u} - \mathbf{v}_i|_1 \Bigg\} \end{aligned} \tag{4.9}$$

results in an equivalent saddle point formulation of type Eq. 4.6 with

$$\begin{aligned}
\mathbf{y} &= \begin{pmatrix} \mathbf{p} \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_K \end{pmatrix} \\
G(\mathbf{x}) = G(\mathbf{u}) &= -\sum_{i=1}^{S} \lambda_i^{(p)} |\mathbf{u} - \mathbf{v}_i|_1 \\
L &= \begin{pmatrix} \nabla_x \\ \nabla_y \\ \lambda_1^{(d)} \mathrm{diag}(\mathbf{C}_1') \\ \vdots \\ \lambda_K^{(d)} \mathrm{diag}(\mathbf{C}_K') \end{pmatrix} \\
F_{\mathbf{p}}^*(\mathbf{p}) &= I_{|\mathbf{p}|\leq 1} + \frac{\epsilon}{2}|\mathbf{p}|^2 \\
F_{\mathbf{q}}^*(\mathbf{q}_i) &= I_{|\mathbf{q}_i|\leq 1} + \left\langle \lambda_i^{(d)} \hat{\mathbf{C}}_i, \mathbf{q}_i \right\rangle
\end{aligned} \tag{4.10}$$

The energy minimization of the transformed saddle point problem Eq. 4.9 can now be efficiently solved by the primal-dual algorithm [22]

$$\begin{aligned}
\mathbf{y}^{n+1} &= (I + \sigma \partial F^*)^{-1}(\mathbf{y}^n + \sigma L \bar{\mathbf{x}}^n) \\
\mathbf{x}^{n+1} &= (I + \tau \partial G)^{-1}(\mathbf{x}^n - \tau L^T \mathbf{y}^{n+1}) \\
\bar{\mathbf{x}}^{n+1} &= 2\mathbf{x}^{n+1} - \mathbf{x}^n
\end{aligned} \tag{4.11}$$

With the corresponding proximal mappings of the dual variables given by

$$\begin{aligned}
(I + \sigma \partial F_{\mathbf{p}}^*)^{-1}(\mathbf{p}) &= \Pi_{\mathbf{p}}(\mathbf{p}) \\
(I + \sigma \partial F_{\mathbf{q}}^*)^{-1}(\mathbf{q}_i) &= \Pi_{\mathbf{q}}(\mathbf{q}_i - \sigma \lambda_i^{(d)} \hat{\mathbf{C}}_i)
\end{aligned} \tag{4.12}$$

with Π being the elementwise projections onto the corresponding n-dimensional unit balls $\Pi(p) = \frac{p}{\max(1,|p|)}$. Instead of dualizing the data term and therefore adding dual variables to the optimization framework, slowing down the convergence process, we apply the work of [65], who proposed a closed-form solution for minimization problems of the above type $\arg\min_{\mathbf{u}} \sum_i^S \lambda_i^{(p)} |\mathbf{u} - \mathbf{v}_i|$ based on the shrinkage operator in the case of $S = 1 \rightarrow$ the so-called generalized shrinkage operator

$$(I + \tau \partial G)^{-1}(\mathbf{u}) = \mathrm{median}\{\mathbf{v}_1, .., \mathbf{v}_S, \mathbf{h}_0, \mathbf{h}_1, .., \mathbf{h}_S\} \tag{4.13}$$

with

$$\begin{aligned} \mathbf{h}_i &= \mathbf{u} + \tau \mathbf{W}_i \\ \mathbf{W}_i &= -\sum_{j=1}^{i} \lambda_j^{(p)} \; + \; \sum_{j=i+1}^{S} \lambda_j^{(p)} \quad , \quad i = 0, .., S \end{aligned} \tag{4.14}$$

For $S = 1$ this proximal mapping equals the standard shrinkage operator

$$(I + \tau \partial G)^{-1}(\mathbf{u}) = \begin{cases} \mathbf{u} - \tau\lambda & \text{if } \mathbf{u} > \tau\lambda \\ \mathbf{u} + \tau\lambda & \text{if } \mathbf{u} < -\tau\lambda \\ 0 & \text{if } |\mathbf{u}| < \tau\lambda \end{cases} \tag{4.15}$$

Instead of choosing global stepsizes based on the operator norm of L, which would result in slow convergence due to the non-uniform structure of L, the stepsizes of the gradient ascents/descents above are implemented via diagonal preconditioning matrices $\sigma \Rightarrow \Sigma, \tau \Rightarrow T$ as proposed by [87]

$$\begin{aligned} &\Sigma_{j,j} = \frac{1}{\sum_{i=1}^{M} |L_{i,j}|} \; , \; T_{i,i} = \frac{1}{\sum_{j=1}^{N} |L_{i,j}|} \\ &\forall j \in \{1, .., N\}, i \in \{1, .., M\} \end{aligned} \tag{4.16}$$

Remarks: Instead of choosing TV-Huber regularization as a prior model of our 3D scene, one can easily replace the corresponding parts in the optimization scheme by more sophisticated regularizer like TGV [14] or Minimal Surface Regularization [35] for even better capturing non-fronto parallel surfaces. However in our application we do not find this necessary and avoid the additional computational overhead and additional parameter dependences. Depth priors can originate in various sensor coordinate systems (e.g. laser scanners, RGB-D sensors, sparse 3D feature reconstruction) and must be projected to the reference camera frame. Typically a large number of pixel in these projected depth priors are unfilled (having unknown depth) and in these cases the corresponding data term $\lambda^{(p)}$ is set to zero. Also note that while the complete formulation looks cumbersome, broken into the single components their computation is very easy and it is straight forward to add additional constraints or regularizers.

For given images $I_{ref}, I_1, .., I_K$ and corresponding cameras $C_{ref}, C_1, .., C_K$ the complete coarse-to-fine reconstruction algorithm is listed in 4.1.

for s = start_scale : num_scales (from coarse to fine)
  if s > 1
    Upsample $(\mathbf{x}, \mathbf{y})^{(s-1)} \rightarrow (\mathbf{x}, \mathbf{y})^{s}$
  end
  for w=1:nWarps
    Warp all images to the reference frame: $w(I_k^s) = warp(I_k^s, \mathbf{x}^s, C_{ref}^s, C_k^s)$
    Compute image matching costs and derivatives $\mathbf{C}_k(I_{ref}^s, w(I_k^s))$, $\mathbf{C}_k'$
    Compute stepsizes σ, τ based on L and (Eq. 4.16)
    for k=1:nIterPD
      run primal dual step (Eq. 4.11)
    end
  end
end

Algorithm 4.1: Algorithm for warping based 3D reconstruction

Reconstruction of Image Sequences

Instead of computing the coarse-to-fine 3D reconstruction for every frame from scratch, we make use of the temporal coherence of the given image sequences. When a solution of the primal-dual algorithm Eq. 4.11 $(\mathbf{x}, \mathbf{y})_t$ at time step t was computed, we propagate it to the next time step $t + 1$, using forward warping and the relative pose between the camera poses $C_t \rightarrow C_{t+1}$ obtained from the camera tracking step. This warped primal-dual solution $\omega((\mathbf{x}, \mathbf{y})_t, C_t, C_{t+1})$ is then resampled to a target level l of the scaling pyramid and used as initialization of the stereo reconstruction process at time step $t + 1$, starting not from the coarsest level but from level l. Using this propagation scheme, it is sufficient to run the coarse-to-fine reconstruction scheme only on the the finest levels. Based on experiments with camera framerates of 10-30 fps, we only do the pre-initialized reconstruction step on the the finest scale.

Note that because at this point we only have the depth information of time step t, we are restricted to a forward warping approach and therefore have to interpolate invalid pixels after the propagation step. Also note that we cannot simply do an image warping of the primal variable $\mathbf{x}$ but have to transform its depth parametrization from camera frame C_t to C_{t+1}. The drawback of this speedup is obviously the inability to capture large pixel translations (corresponding to fast camera motion in static scenes) by only considering the finest image scales, but for typical image framerates of 10-30 fps and corresponding small inter-frame distances we didn't observe any problems. For large camera displacements, the pyramid simply needs to be traversed completely again.

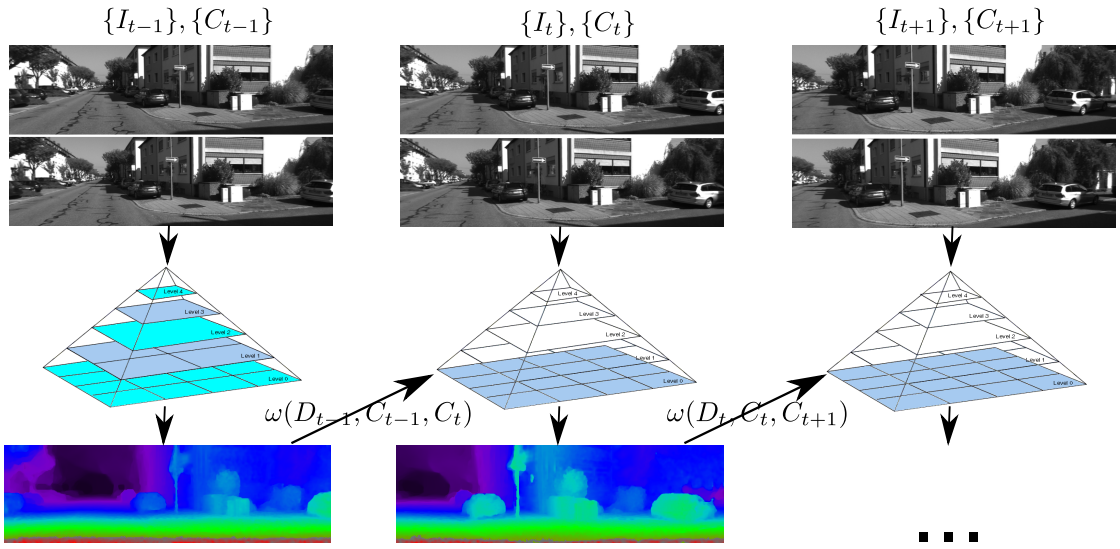


Figure 4.5: 3D reconstruction from a successive image sequence. First depth map is initialized with an arbitrary depth estimate and a full coarse-to-fine warping scheme is applied. For successive stereo frames, the primal-dual algorithm is initialized with the warped results from the previous frame and optimization is done only on the original image scale. Sparse depth prior can be included in the optimization as well at any time.

Large-Scale Dense SLAM

The tracking of the camera positions is performed by direct image alignment robustified to global affine illumination changes, with an additional check for loop-closures resulting in a pose-graph optimization on the constraints between keyframe poses. Tracking and pose refinement are implemented using the publicly available implementation of [28].

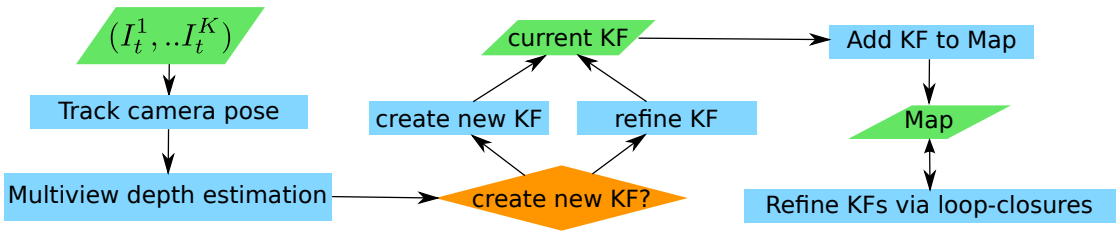


Figure 4.6: Flowchart of the SLAM system

Results

For efficiency reasons the variational dense reconstruction algorithms have been implemented on GPU, utilizing the massively parallel architecture available nowadays on even modest hardware. All algorithms are quite robust w.r.t. their parameters such that no delicate tuning is required. The main parameter to tune is $\lambda^{(d)}$ – the weighting of the data term vs. the regularizer. In all our experiments we set $\lambda^{(d)} = 2.0$, $\lambda^{(f)} = 0.5\lambda^{(d)}$ and for the Huber regularizer $\epsilon = 1e-4$. For the reconstruction pyramid we set a scaling factor of 0.5, 5 warps per scale level and 50 primal-dual iterations per warp.
For the qualitative evaluation in Sec. 4.5.3 we used the sparse laser data provided by the KITTI odometry dataset as depth priors, weighted equally as the image matching data terms ($\lambda^{(p)} = \lambda^{(d)}$).

SLAM - KITTI Odometry Benchmark

For evaluation we used the KITTI odometry dataset [34], containing 21 automotive sequences performing different driving scenarios with about 42,000 stereo image pairs in total, recorded at 10 fps. As in [29] we ran the reconstruction on half-size resolution (620×184) to achieve a good trade-off between accuracy and speed. In contrast to our expectations, the resulting pose accuracy did not improve w.r.t. our baseline algorithm of LSD-SLAM [28], mainly in scenes containing dynamic objects. The direct image alignment then could not cope anymore with the amount of outliers represented by the pixel values belonging to non-static scene elements. Therefore we resorted to only use the sparse feature reconstruction for camera tracking (less pixels with better accuracy), while the dense reconstruction is propagated to the keyframes and the corresponding 3D point cloud. As described in section 4.3.1 we use the sparsely reconstructed features as depth prior for our dense reconstruction. This somewhat intuitive finding – that less but more accurately determined 3D points are used for pose estimation instead of many less accurate 3D points – was evaluated systematically very recently in the work of [27].

We processed the KITTI visual odometry dataset on a standard notebook containing an Intel i5-5200U CPU at 2.20GHz and a NVidia GeForce 830M GPU. The timing is dominated by the three main parts of our SLAM system – tracking, mapping and constraint search for the pose-graph optimization. Results are given in table 4.1, yielding an average processing rate of 14 frames per second on a standard notebook. The bottleneck clearly is the dense stereo reconstruction. However we would like to point out that the corresponding GPU (GeForce 830M) has an official processing power of only 527GFlops, whereas current highend GPUs in the consumer market reach up to 10,974 GFlops (NVidia Titan X). Running our system on a moderate GPU in between (NVidia GTX 970; 3,494 GFlops), the dense stereo

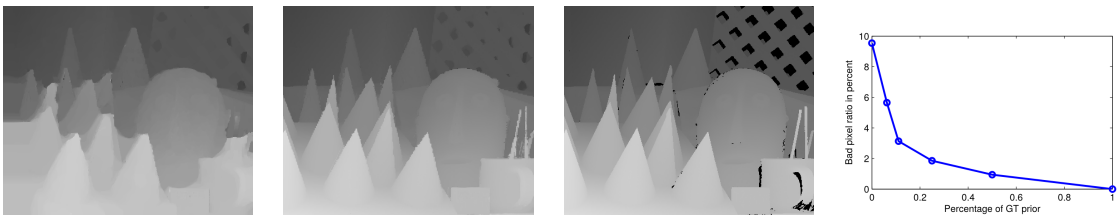
	Average time per frame
Tracking	11ms
Dense Mapping	45ms
Pose-Graph	17ms

Table 4.1: Averaged timings per frame over the complete KITTI visual odometry benchmark test dataset (620 × 184, 20k stereo image pairs) - using a commodity notebook system

reconstruction has an average runtime of 12ms (≈ 80 fps). Additionally note that the encasing SLAM framework of [28] runs solely on CPU, whereas the dense stereo reconstruction nearly solely uses the GPU – favoring a threaded implementation.

Sparse Priors

To quantitatively show the effects of adding sparse prior depth information early into the stereo reconstruction process, we provide an experiment in Fig. 4.7, where we used the ground truth information provided by the Middlebury stereo dataset [99] as additional depth prior. When using the complete ground truth, the bad pixel ratio of the resulting disparity map is at 0% as expected and rising, if a lesser fraction of that ground truth is taken into account (see Fig. 4.7). Real life data however can contain outliers and noise, leading to false depth priors. As we however weight the influence of these depth priors only linearly using the L_1 norm and employing a smoothness term additionally (see Eq. 4.5) the overall minimization is somewhat robust to gross outliers.

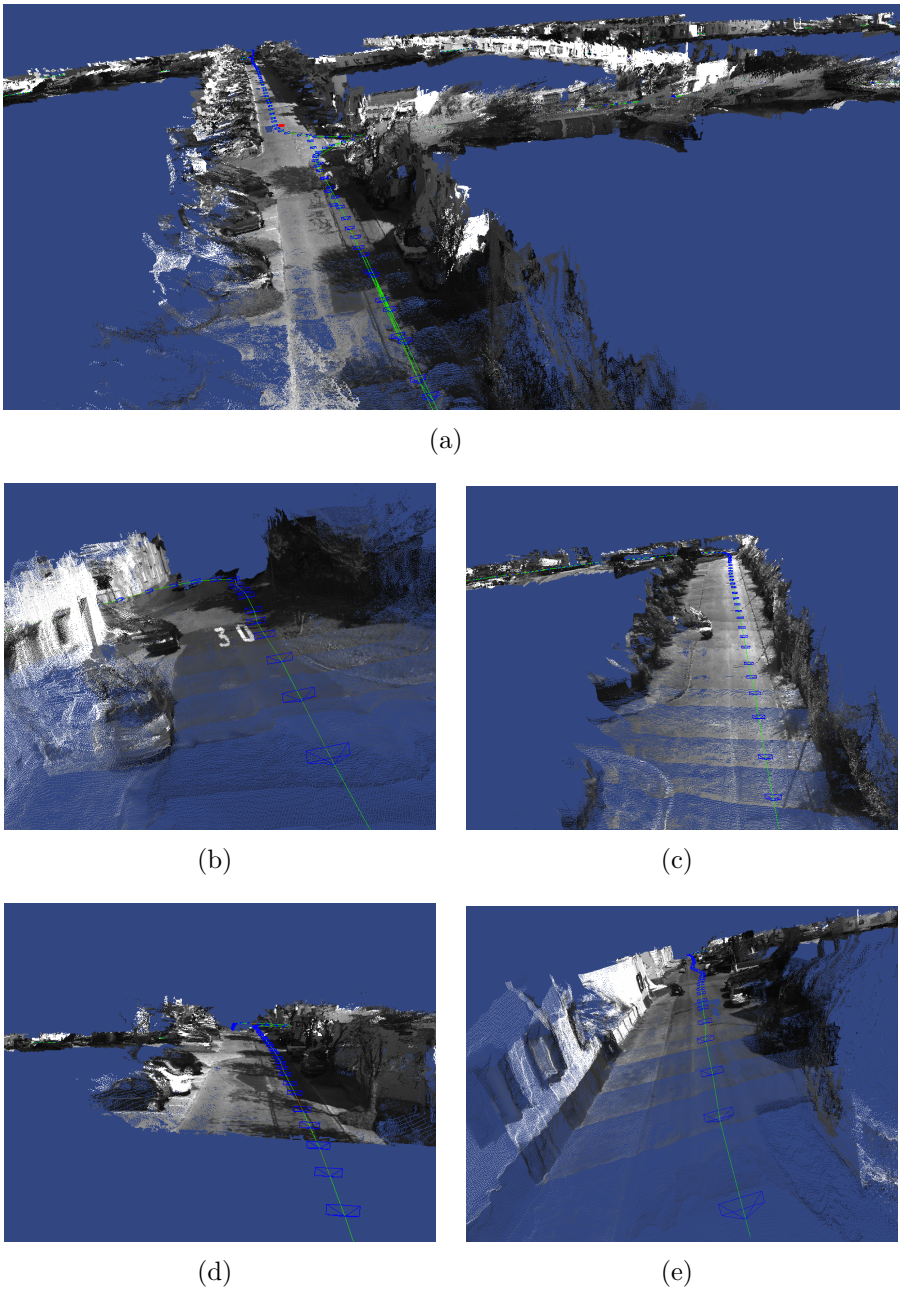


(a) No priors (b) Ground truth as priors (25%) (c) Ground truth (d) Bad pixel ratios

Figure 4.7: Stereo reconstruction results on Middlebury data with different density of sparse depth priors. (d): Bad pixel ratios for stereo reconstruction results using different subsets of the ground truth (GT) as sparse priors

Qualitative Results

In Fig. 4.8 we show some qualitative results of the estimated point clouds taken from the same dataset as Fig. 4.1.



(a) (b) (c) (d) (e)

Figure 4.8: Dense large-scale reconstruction using automotive image data (KITTI sequence 00) of 4,500 stereo image pairs.

Conclusions

We presented a direct visual SLAM system which recovers dense large scale environments from color videos in real-time on a commodity notebook. It combines a direct image alignment and pose graph optimization of keyframes for globally consistent camera tracking with a variational approach for dense reconstruction of the keyframes. The latter combines an L1-integration of multiple input images and optional depth priors with a spatial regularization imposing structural smoothness of the scene geometry.

In addition, the proposed approach can easily integrate other depth measurements, for example from a laser scanner. Experimental results on the KITTI dataset demonstrate that we can robustly recover large-scale dense city maps from a stereo video in real-time. We believe such real-time dense reconstructions will form a vital ingredient for self-driving cars and autonomous robots as they are a basis for obstacle avoidance and path planning.

Chapter 5

Depth Map Fusion

Summary

In this work we propose an algorithm for robustly fusing digital surface models (DSM) with different ground sampling distances and confidences, using explicit surface priors to obtain locally smooth surface models. Robust fusion of the DSMs is achieved by minimizing the L1-distance of each pixel of the solution to each input DSM. This approach is similar to a pixel-wise median and most outliers are discarded. We further incorporate local planarity assumption as an additional constraint to the optimization problem, thus reducing the noise compared to pixel-wise approaches. The optimization is also inherently able to include weights for the input data, therefore allowing to easily integrate invalid areas, fuse multi-resolution DSMs and to weight the input data. The complete optimization problem is constructed as a variational optimization problem with a convex energy functional, such that the solution is guaranteed to converge towards the global energy minimum. An efficient solver is presented to solve the optimization in reasonable time, e.g. running in real-time on standard computer vision camera images. The accuracy of the algorithms and the quality of the resulting fused surface models is evaluated using synthetic datasets and spaceborne datasets from different optical satellite sensors. [1]

Contributions
The main author did all the following theoretical and implementation work on his own, whilst the evaluation part was done together with David Gaudrie, plus conceptual discussions with the co-authors:

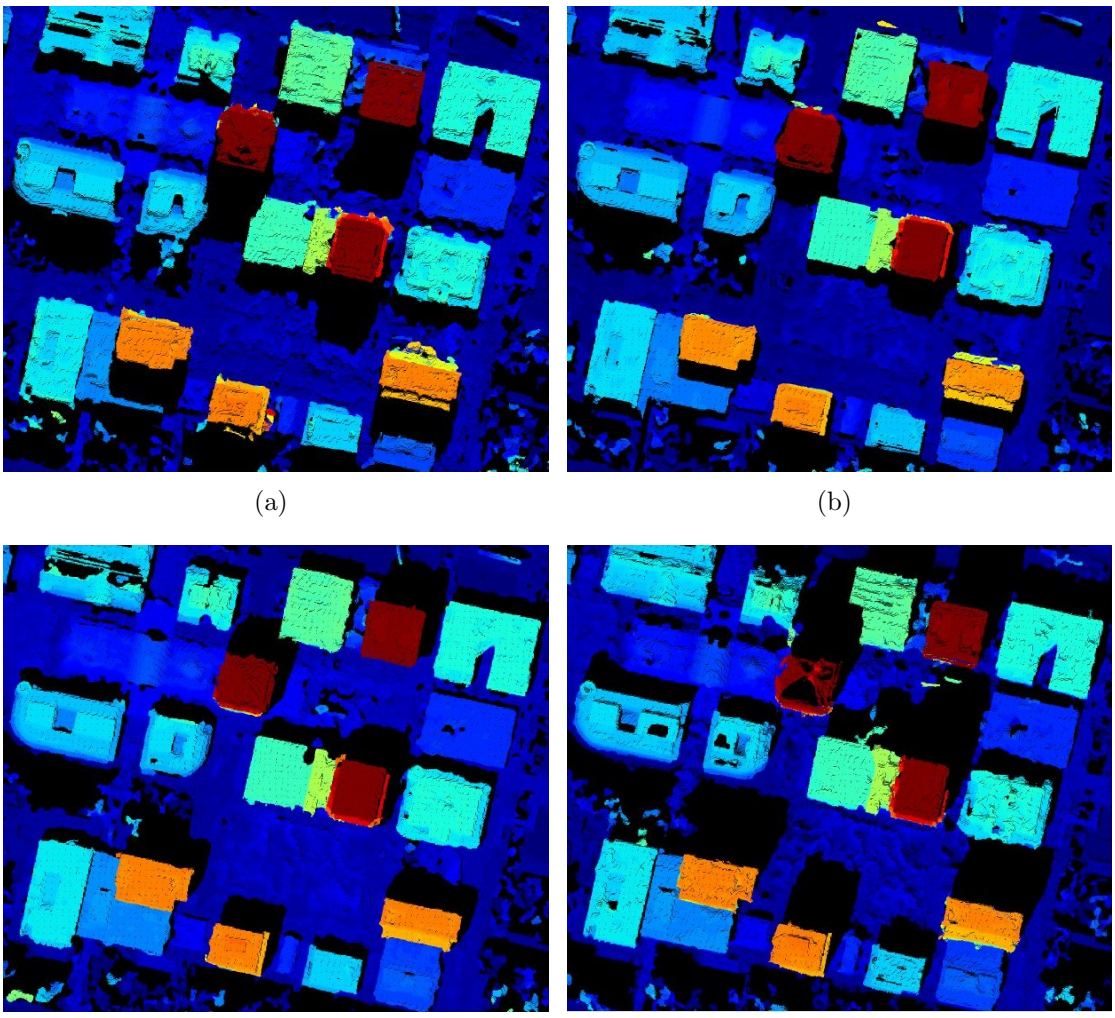
- Multi-resolution fusion of large DSMs using second order smoothness-priors
- Adaptive weighting of input data of differenct accuracy and missing data

[1]

Introduction

With an ever increasing amount of earth observation sensors, the problem of having data at all, increasingly shifts towards the problem of how to make best use of an abundance of data. One aspect of remote sensing data is the 3D information contained in the observed images, resulting in digital surface models (DSM), constituting a basic component for many applications, such as orthophoto creation, mapping, visualization and 3D planning. As many technologies for DSM generation exist (airborne LiDAR, SAR interferometry, automatic image matching, ..) the corresponding results differ in their characteristics and quality in general. Because of the decreasing revisit time for many parts of the Earth's landmass, multiple datasets of DSMs are available for these regions and it is therefore interesting to fuse these into a single DSM with higher accuracy. Depending on the underlying satellite characteristics like ground sampling distance (GSD), the DSMs capture different parts of the scene in different quality, which even can be mutually exclusive to some extent. For example, high resolution sensors like WorldView-2 with a GSD of 0.5m perform very well in urban areas, whereas the results in forest areas are somewhat moderate. In contrast, Cartosat-1 with a GSD of 2.5m performs quite opposite in these areas [108]. Even with the same sensor, a different exposure time can drastically alter the results in shadow areas or in highly reflective areas like glaciers. Clouds are posing an additional problem for optical image sensing, providing no valid data in these areas, thereby requiring these gaps to be filled in by valid data from other sensors or another timestamp. A prominent example for a large data abundance is aerial imaging, which typically produces large image streams with image overlaps >80%. For computing the corresponding 3D reconstruction, many multi-view image matching techniques match stereo image pairs individually and later fuse the resulting DSMs into a common height model, see e.g. [46], [56], [97].
Our work focuses on the fusion of 2.5D DSM grids, with a resolution from several decimeters to a few meters. We use the common notation of 2.5D to explicitly distinguish between 3D point cloud registration / fusion and fusing their projections in a common 2D reference frame. The latter consists of 2D images, each pixel containing its height above ground and is commonly referred as 2.5D DSM, as it contains 3D height information but not to full extent (e.g. no bridges can be modelled). DSM fusion has been considered by various authors previously. The simplest method is based on weighted averaging of two or more height maps [103], [92]. As weighted averaging cannot deal with outliers or blunders in the DSMs, a median fusion is often used for multi-DSM fusion, sometimes followed by weighted averaging of the inliers [46]. Both median and weighted averaging process each pixel independently, and thus cannot take into account the local surface shape, which is regular for many areas. Applying additional mean or median based filtering spatially reduces the amount of noise to some extent, at the cost of blurring potentially sharp edges. An

example for context aware fusion algorithms is the use of sparse representations [78], where a DSM patch is computed as a sparse linear combination of dictionary DSM patches. Except for median fusion, pixel-wise error maps are required by weighted averaging and sparse representations. A comparison between weighted averaging and sparse representations [101] found that the quality of the fused DSMs is mostly determined by the quality of these pixel based error maps.



(a) (b)

(c) (d)

Figure 5.1: (a)-(d): Four co-registered DSMs, obtained from optical stereo reconstruction using [59] for different camera view points / satellite positions (noticeable by the different invalid occlusion areas in black).

Another direction of work aims at formulating a global energy functional, minimizing the distance of the fused result to all input DSMs simultaneously and additionally incorporating the assumption of the world being locally planar

([127, 126, 87, 80]). Due to its simple structure and theoretically well founded minimization procedure, we build upon this work and extend it to a weighted, multi-resolution, fusion framework.

Method

As basic fusion algorithms we are looking at the following pixel-wise fusion methods: mean and median fusion, as well as *medmean fusion*. We define the latter one as median based fusion that reduces the amount of outliers in the fused DSM by averaging the median value for each pixel with all other DSM heights of this pixel being at a distance of less than 2m from the median value. Note that this is an empirical threshold, depending on the overall height range and noise level. In contrast to these simple pixel-wise fusion methods, advanced methods usually enforce some kind of spatial smoothness constraint to get closer to a physical meaningful solution, with neighboring image pixels forced to have a similar height value. Note that this constraint often is in contrast to the data term (height values) of the involved images, where neighboring pixels can differ significantly in height. This leads to the general formulation of our DSM fusion problem as

$$\min_{\mathbf{u}} \left\{ R(\mathbf{u}) + \lambda_d \sum_{k=1}^{K} \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{5.1}$$

where $\mathbf{u} \in \mathbb{R}^{M \cdot N}$ is the *'optimal'* DSM to solve for, already written as stacked vector of pixels to simplify notation in the following. The K (noisy) input DSMs are given as $\mathbf{g_k}$ (see e.g. Figure 5.1), the scalar factor λ_d is balancing the impact of the smoothness term and the data term and $R(u)$ is depicting a general regularizer on u.

In the case of DSM fusion, these smoothness constraints (or regularizers) are the assumption of the world being locally planar, meaning that the height value of each pixel of the DSM depends on its local context and e.g. is highly unlikely to have a significantly different height value than its surrounding pixels.

This smoothness constraint typically is implemented by minimizing the sum of gradients of the resulting DSM in both x and y-direction, resulting in large partial differential equation systems (PDE).

In recent years, Total Variation based methods (TV) for minimizing energy functionals have seen a lot of attention in the research community. One reason is that these algorithms are very well-suited for parallelization and, together with the recent advances of GPU-based computational power, lead to efficient algorithms, solving these optimization problems efficiently. And as the energy functional of our image fusion problem is written in a convex formulation, the solution is globally optimal and independent of its initialization. Since the second term of Equation 5.1 is always

convex in the variable $\mathbf{u}$ to solve for (sum of norms), the complete energy functional is convex, if the regularizer $R(\mathbf{u})$ is convex. The two regularizers used in this paper are described in Section 5.3.1 (namely TV and TGV) and are simply linear transformations of the type $K \cdot \mathbf{u}$. Therefore throughout this paper Equation 5.1 will always be convex.

TV-L_1 Fusion

Based upon the Rudin-Osher-Fatemi image denoising model (ROF-model) [96], the extension for multiple image fusion, together with replacing the quadratic data term by the more robust L_1 norm as in [87] is written as

$$\min_{\mathbf{u}} \left\{ \|\nabla \mathbf{u}\|_1 + \lambda_d \sum_{k=1}^{K} \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{5.2}$$

Note that the choice of the L_1 norm for both the gradient and the data term plays an important role for the fusion of multiple noisy DSMs (or images in general) for the following reasons: Applied to the regularizer (gradient) it still enables the solution to exhibit strong edges / discontinuities (e.g. at the transition of house roof tops to street level), as these height value jumps are only penalized linearly. Applying the L_1 norm to the second term - the data term - makes the whole fusion process robust to outliers as well, as these also are only weighted linearly in the optimization process and their influence therefore is limited compared to *e.g.* a least squares minimization approach. While this model already provides good results by smoothing flat areas and preserving sharp discontinuities, it suffers from the so-called staircasing effect. This effect is a direct result of the regularizer, whose assumption is a locally planar world - where planar unfortunately refers to locally fronto-parallel. This staircasing effect of the TV-L_1 algorithm is visible in Figure 5.2(f), resulting in a slanted roof which is not smooth. One way to overcome this issue is using the Huber norm instead of the pure L_1 norm for the regularizer, thereby penalizing small height differences quadratically and larger difference as before using the L_1 norm. This results in a locally more smooth surface, mitigating the staircasing effect to some extent. The authors of [87] added this Huber regularized fusion method as one baseline method to compare their algorithms against. However, this does not solve the issue of reconstructing large non-fronto-parallel surfaces (slanted planes). To achieve that goal, a more advanced smoothness assumption as in the following section is required. For further details about the results of TV-Huber-based regularization, we refer to the work of [87].

TGV-L_1 Fusion

To overcome the fronto-parallel assumption of TV-L_1 minimization, [14] introduced the mathematical model of Total Generalized Variation (TGV) has been introduced

as a higher-order extension of Total Variation which favors the solution to consist of piecewise polynomial functions (e.g. fronto-parallel, affine, quadratic). Especially the 2nd order is of high interest, as it forces the solution to consist of piecewise planar functions, which means that compared to the fronto-parallel TV-L_1 model, the regularizer now also favors slanted planes. [87] applied this model to DSM fusion, resulting in the following optimization problem

$$\min_{\mathbf{u},\mathbf{v}} \left\{ \lambda_s \|\nabla_u \mathbf{u} - \mathbf{v}\|_1 + \lambda_a \|\nabla_v \mathbf{v}\|_1 + \lambda_d \sum_{k=1}^{K} \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{5.3}$$

Now, before the variation of the image $\mathbf{u}$ is measured, a 2D vector field $\mathbf{v}$ is subtracted from the gradient of $\mathbf{u}$. An affine surface in the image $\mathbf{u}$ has a constant gradient $\nabla\mathbf{u}$, so by coupling and minimizing $|\nabla\mathbf{u} - \mathbf{v}|$, the vector field $\mathbf{v}$ will also be constant and it's gradient $\nabla\mathbf{v}$ therefore zero. Regarding our overall optimization problem, this means that the energy term will be lower, if affine functions can be found in the image, whereas non-affine functions get additional penalties by $|\nabla\mathbf{v}|$. The values $\lambda_s, \lambda_a, \lambda_d$ are scalar weights and balance the impact of the smoothness term, the affine term and the data term. Note that we now notationally need to differ between two gradient operators, $\nabla_u \in \mathbb{R}^{MN\times 2MN}$ and $\nabla_v \in \mathbb{R}^{2MN\times 2MN}$ as the corresponding vector spaces are of different dimension (see Section 5.4.1).

Weighted TGV-L_1 Fusion

When fusing DSMs it is desirable to weight the input DSMs on a per pixel base, to be able to incorporate additional prior knowledge into the fusion process. This prior knowledge for example can be based on the different sensor characteristics used to generate the DSM, confidence measures during the 3D reconstruction process itself, information about occluded and therefore unknown areas in each DSM, etc. We therefore extend Equation 5.3 with a weighting matrix W_k for each input DSM

$$\min_{\mathbf{u},\mathbf{v}} \left\{ \lambda_s \|\nabla\mathbf{u} - \mathbf{v}\|_1 + \lambda_a \|\nabla\mathbf{v}\|_1 + \lambda_d \sum_{k=1}^{K} W_k \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{5.4}$$

Parameters

This optimization problem (and the ones in Equation 5.2 and 5.3) is very parameter dependent, as we need to adapt the influence of the data term λ_d manually for datasets with different ranges of $g_k^{(i,j)} \in \mathbf{g_k}$ as well as for a different number K of input images. To achieve independence of the data range of the input DSMs, we scale all input data to the interval [0..1]

$$g_k^{(i,j)} = \frac{g_k^{(i,j)} - g_{min}}{g_{max} - g_{min}} \tag{5.5}$$

with $g_{min} = \min_{i,j,k} g_k^{(i,j)}$ and $g_{max} = \max_{i,j,k} g_k^{(i,j)}$. The independence from K is achieved by normalizing the influence of the data term w.r.t. the two-image case and using the adaptive

$$\lambda_d^K = \frac{2}{K}\lambda_d \tag{5.6}$$

Note that we do not need all 3 weighting factors $\lambda_s, \lambda_a, \lambda_d$, as we can multiply the whole energy functional with $\frac{1}{\lambda_d}$. We therefore only have to deal with λ_s, λ_a and $\lambda_d = 1$ implicitly. Additionally it is a good choice to set $\lambda_a = 4\lambda_s$, which leaves us with only one parameter λ_s to choose between a large smoothing impact ($\lambda_s >>$) or a more data-driven fusion ($\lambda_s <<$). Choosing λ_a too big results in oversmoothing of discontinuities – we loose some of our edge-preserving capability. When choosing λ_a very small, we obtain results closer to pure TV-L_1 (together with the staircasing effects). To avoid an additional free parameter, we coupled the value to the smoothness weighting λ_s and experimented with different correlation factors. In all our empirical tests over different artificial and natural datasets the choice $\lambda_a = 4\lambda_s$ produced consistently good results.

All these extensions and modifications apply to the TV-L_1 method similarly. In the next section we will go into detail about how to solve these optimization problems numerically.

Optimization

In the following we describe the numerical optimization of our weighted TGV-L_1 energy functional given in Equation 5.4. The solution for the TV-L_1 energy functional is similar and can be derived easily from the solution below.
To solve for the fused DSM $\mathbf{u} \in \mathbb{R}^{M\times N}$ (in the following written as stacked vector $\mathbb{R}^{MN\times 1}$) in Equation 5.4, we need to overcome the non-differentiable L_1-norm, which complicates any gradient descent based minimization scheme. An efficient algorithm which elegantly circumvents the differentiability problem of the gradient operator is the primal-dual algorithm of [22]. By applying the Legendre-Fenchel transform to the terms involving the derivative of the primal variables we obtain the dual formulation / conjugate of these terms as

$$\begin{aligned} \lambda_s\|\nabla\mathbf{u} - \mathbf{v}\|_1 &= \max_{\mathbf{p}\in P}\{\langle\nabla\mathbf{u} - \mathbf{v}, \mathbf{p}\rangle\} \\ \lambda_a\|\nabla\mathbf{v}\|_1 &= \max_{\mathbf{q}\in Q}\{\langle\nabla\mathbf{v}, \mathbf{q}\rangle\} \end{aligned} \tag{5.7}$$

such that the transformed saddle-point problem of Equation 5.4 in the primal variables $\mathbf{u}, \mathbf{v}$ and the dual variables $\mathbf{p}, \mathbf{q}$ with constraints

$$\begin{aligned} P &= \{\mathbf{p} \in \mathbb{R}^{2MN} : \|\mathbf{p}\|_\infty \leq \lambda_s\} \\ Q &= \{\mathbf{q} \in \mathbb{R}^{4MN} : \|\mathbf{q}\|_\infty \leq \lambda_a\} \end{aligned} \tag{5.8}$$

is

$$\min_{\mathbf{u},\mathbf{v}} \max_{\mathbf{p},\mathbf{q}} \left\{ \langle \nabla \mathbf{u} - \mathbf{v}, \mathbf{p} \rangle + \langle \nabla \mathbf{v}, \mathbf{q} \rangle + \lambda_d \sum_{k=1}^{K} W_k \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{5.9}$$

A detailed explanation of the dual variables and the corresponding vector spaces is given in Section 5.4.1. With the convex saddle-point problem above (Equation 5.9), we can now directly apply the primal-dual algorithm of [22] to get the following optimization scheme, which is basically iteratively performing gradient descents on the primal variables and gradient ascents on the dual variables:

Input: $\mathbf{u}^0, \mathbf{v}^0, \mathbf{p}^0, \mathbf{q}^0 = \mathbf{0}$, $\bar{\mathbf{u}}^0 = \mathbf{u}^0, \bar{\mathbf{v}}^0 = \mathbf{v}^0$, $\theta = 1$, step sizes $\tau_i > 0$
Iterations $n \geq 0$:

$$\begin{cases} \mathbf{p}^{n+1} = \Pi_P \left(\mathbf{p}^n + \tau_p \lambda_s (\nabla \bar{\mathbf{u}}^n - \bar{\mathbf{v}}^n) \right) \\ \mathbf{q}^{n+1} = \Pi_Q \left(\mathbf{q}^n + \tau_q \lambda_a (\nabla \bar{\mathbf{v}}^n) \right) \\ \mathbf{u}^{n+1} = \text{prox}_f \left(\mathbf{u}^n + \tau_u \lambda_s \nabla^* \mathbf{p}^{n+1} \right) \\ \mathbf{v}^{n+1} = \mathbf{v}^n + \tau_v (\lambda_a \nabla^* \mathbf{q}^{n+1} + \lambda_s \mathbf{p}^{n+1}) \\ \bar{\mathbf{u}}^{n+1} = \mathbf{u}^{n+1} + \theta (\mathbf{u}^{n+1} - \mathbf{u}^n) \\ \bar{\mathbf{v}}^{n+1} = \mathbf{v}^{n+1} + \theta (\mathbf{v}^{n+1} - \mathbf{v}^n) \end{cases}$$

Algorithm 5.1: Primal-dual optimization algorithm for TGV-L1-based image fusion

For details about the linear operators ∇ and their negative adoints ∇^*, as well as the step sizes τ_i for the gradient descents see Section 5.4.1. To ensure the constraints of Equation 5.8, the corresponding proximal mappings of the dual variables are given as simple point-wise projections

$$\begin{aligned} \Pi_P(\mathbf{p}) &= \frac{\mathbf{p}}{\max\{1, \|\mathbf{p}\|/\lambda_s\}} \\ \Pi_Q(\mathbf{q}) &= \frac{\mathbf{q}}{\max\{1, \|\mathbf{q}\|/\lambda_a\}} \end{aligned} \tag{5.10}$$

The proximal mapping of the primal variable u, enforcing the data constraints $\min \sum_k \|u - g_k\|$ is slightly more complicated. In previous work, [87] and [60] added Lagrange multipliers for each observation ($\langle r_k, u - g_k \rangle$) and optimized the energy functional with an additional gradient descent scheme for these auxiliary variables. Here we build upon the work of [65] to solve this constraint exactly and directly, thus avoiding an additional iterative scheme. We therefore don't need further dual variables for every observation as in [87], resulting in less memory consumption. As the closed-form solution of the proximal mapping is computationally simple, it further results in a noticeable speedup compared to solving it via an iterative

gradient-descent based primal-dual scheme. Defining

$$f(x) = \lambda\tau \sum_{k=1}^{K} w(x,k) \cdot \|x - g_k\|_1 \tag{5.11}$$

the proximal mapping is given as

$$\text{prox}_f(x) = \arg\min_y \left\{ \frac{1}{2}\|x-y\|_2^2 + \lambda\tau \sum_{k=1}^{K} w(y,k) \cdot \|y - g_k\|_1 \right\}$$

whose solution is given by a generalized shrinkage formula according to [65]:

$$\text{prox}_f(x) = \text{median}\{g_1, ..., g_K, p_0, p_1, ..., p_K\} \tag{5.12}$$

with

$$p_i = x + \tau\lambda W_i \tag{5.13}$$

$$W_i = -\sum_{j=1}^{i} w(x,j) \; + \; \sum_{j=i+1}^{K} w(x,j) \tag{5.14}$$

Implementation Details

For discretization of the gradient operators $\nabla_u : \mathbb{R} \to \mathbb{R}^2$ and $\nabla_v : \mathbb{R}^2 \to \mathbb{R}^4$, we use forward finite differences with Neumann boundary conditions

$$\nabla_u = \begin{pmatrix} \nabla_x \\ \nabla_y \end{pmatrix}, \quad \nabla_v = \begin{pmatrix} \nabla_x & 0 \\ \nabla_y & 0 \\ 0 & \nabla_x \\ 0 & \nabla_y \end{pmatrix}, \quad \nabla_x, \nabla_y \in \mathbb{R}^{MN \times MN} \tag{5.15}$$

where

$$\begin{aligned} (\nabla_x \mathbf{u})_{\gamma(i,j)} &= \begin{cases} \mathbf{u}_{\gamma(i+1,j)} - \mathbf{u}_{\gamma(i,j)} & \text{if } i < M \\ 0 & \text{if } i = M \end{cases} \\ (\nabla_y \mathbf{u})_{\gamma(i,j)} &= \begin{cases} \mathbf{u}_{\gamma(i,j+1)} - \mathbf{u}_{\gamma(i,j)} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases} \end{aligned} \tag{5.16}$$

are the forward finite differences in x and $y-$direction and the function $\gamma : \mathbb{Z}\times\mathbb{Z} \to \mathbb{Z}$ mapping the indices from 2D image space to 1D stacked vector notation

$$\gamma(i,j) = (i-1)M + j \tag{5.17}$$

The corresponding negative adoint operators ∇^*, needed for the gradient descent in the dual variables of Algorithm 5.1, are simply the corresponding transposed and negated matrices $\nabla^* = -\nabla^T$. Note that these are sometimes in literature also referred to as divergence operators. When written explicitly, the above definition naturally reads as backward finite differences with Dirichlet boundary conditions

$$\begin{aligned} \nabla_u^* \mathbf{p} &= -\begin{pmatrix} \nabla_x & \nabla_y \end{pmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \end{pmatrix} \\ (\nabla_u^* \mathbf{p})_{i,j} &= \begin{cases} \mathbf{p}^1_{i,j} - \mathbf{p}^1_{i-1,j} & \text{if } 1 < i < N \\ \mathbf{p}^1_{i,j} & \text{if } i = 1 \\ -\mathbf{p}^1_{i-1,j} & \text{if } i = N \end{cases} \\ &+ \begin{cases} \mathbf{p}^2_{i,j} - \mathbf{p}^2_{i,j-1} & \text{if } 1 < j < M \\ \mathbf{p}^2_{i,j} & \text{if } j = 1 \\ -\mathbf{p}^2_{i,j-1} & \text{if } j = M \end{cases} \end{aligned} \tag{5.18}$$

The implementation is similar for the second operator ∇_v and its negative adjoint. Although the mathematical notation may imply a very large optimization problem (e.g. $\nabla_x \in \mathbb{R}^{MN \times MN}$), the corresponding matrices are very sparse: ∇_u, ∇_v only have two non-zero elements per matrix row. Therefore implementation can be done efficiently either using a sparse matrix representation, or avoiding this overhead by directly computing the gradient and divergence per pixel using Equations 5.16 and 5.18.

To ensure convergence of the primal-dual algorithm, the step sizes of the gradient ascents/descents are bound to the operator norm of the linear operators described in Equation 5.15 according to [22] as follows

$$\tau_u \tau_p \leq \frac{1}{||\nabla_u||^2_{op}} \qquad \text{and} \qquad \tau_v \tau_q \leq \frac{1}{||\nabla_v||^2_{op}} \quad . \tag{5.19}$$

Due to the simple structure of the forward differences the step sizes can be explicitly computed as $\tau_u = \tau_p = \tau_v = \tau_q = 1/\sqrt{8}$.

The whole algorithm stops, if either a predefined maximum number of iterations has been reached or the energy change between successive iterations drops below a relative threshold. Due to the stacked vector notation, the input weights are denoted as diagonal matrices W_k and the corresponding multiplications are actually a pixel-wise multiplication.

Since the algorithm is inherently suited for parallelization, the algorithm was implemented on GPU, allowing for a processing speed of 40ms for 10 images with a size of 640×480 (using a Nvidia GTX 970). Since GPU memory cannot be easily swapped to the harddrive and the DSMs to fuse are usually quite large (near Gigapixel range for satellite data), we process larger data by tiling it into overlapping smaller regions, solving these separately. The overlap is chosen as 5% of the corresponding width of

the tiles, means that for *e.g.* quadratic tiles of 1000 pixel width, the overlap w.r.t. to the neighboring tile amounts to 50 pixel. To further account for the less accurate results at the tile borders, we employ the same strategy as used by [46]. Instead of just computing the mean value of neighboring tiles in the overlapping area, a weighted mean is used, such that the corresponding weights decrease linearly towards the tile border. Of course, when handling such large DSMs and processing them in tiles the overall solution is not globally optimal anymore. The tiling size is computed as large as possible while the complete data still fits into GPU memory. With the memory overhead of TGV-L_1 based optimization and *e.g.* 5 input DSMs, this amounts to tiles of roughly 8000×8000 pixel for a current GPU having 8GB of memory.

Evaluation

Artificial Tests

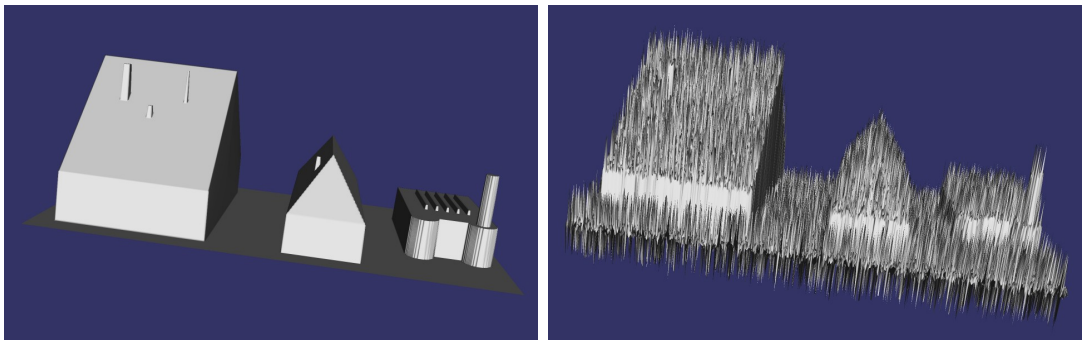
The first evaluation is done on synthetic data. A given ground truth DSM $\mathbf{g}$ with a height range of [0..170] is perturbed with Gaussian noise and with salt and pepper noise to simulate different noisy observations of the scene. Five of these noisy DSMs are then given as input to the fusion algorithms and the accuracy of the output DSM $\mathbf{u}$ is measured by the logarithmic signal-to-noise ratio:

$$SNR = 10 \log_{10} \left(\frac{I^2_{\text{signal}}}{I^2_{\text{noise}}} \right) = 10 \log_{10} \left(\frac{||\mathbf{g}||^2}{||\mathbf{u} - \mathbf{g}||^2} \right) \tag{5.20}$$

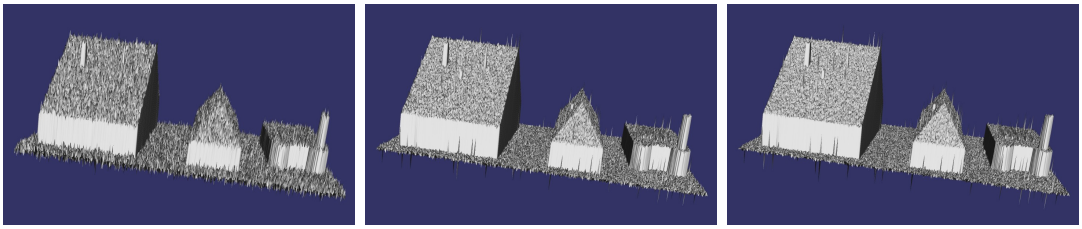
In Figure 5.2, visual and numerical results are given, showing a significantly higher accuracy of the global optimization methods for DSM fusion over simple mean and median based fusion. We can also remark the staircasing effects provided by TV-L_1 fusion resulting in a non-smooth roof in Figure 5.2 (f), as well as the smoothness of TGV-L_1 fusion, which has both the best SNR and the best visual aspect. To obtain a fair comparison between TV-L_1 and TGV-L_1 based fusion, we ran the algorithms for varying λ_d values and chose the parameter which resulted in the highest SNR value – compare Figure 5.3. Furthermore the noise was fixed for the different runs as well.

Artificial Tests - Weights

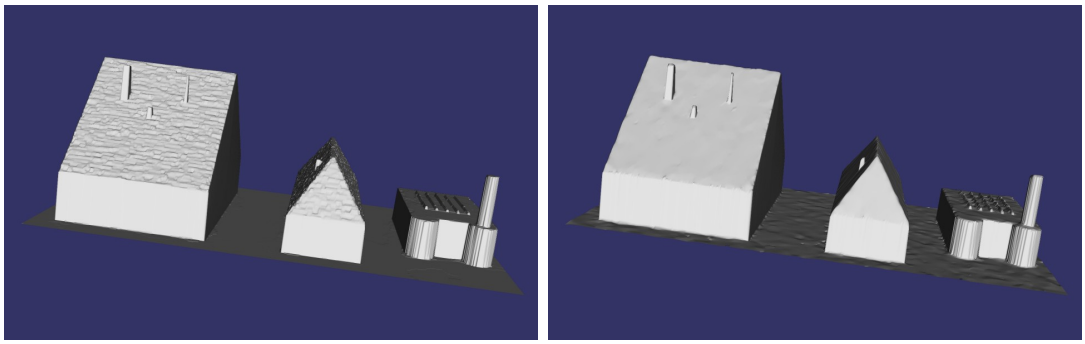
In this experiment, we compare the basic fusion of Equation 5.2 and 5.3 against the formulation using an explicit weighting scheme as proposed in Equation 5.4. To this end, we add a wrong systematic bias to 3 of our 5 input images (compare Figure 5.4 (c) and set corresponding weights $w = 0.2$ for these areas, whereas the rest is set to $w = 1.0$. Note that we deliberately did not set the weights for the wrong



(a) Groundtruth data (b) 1 of 5 noisy input images



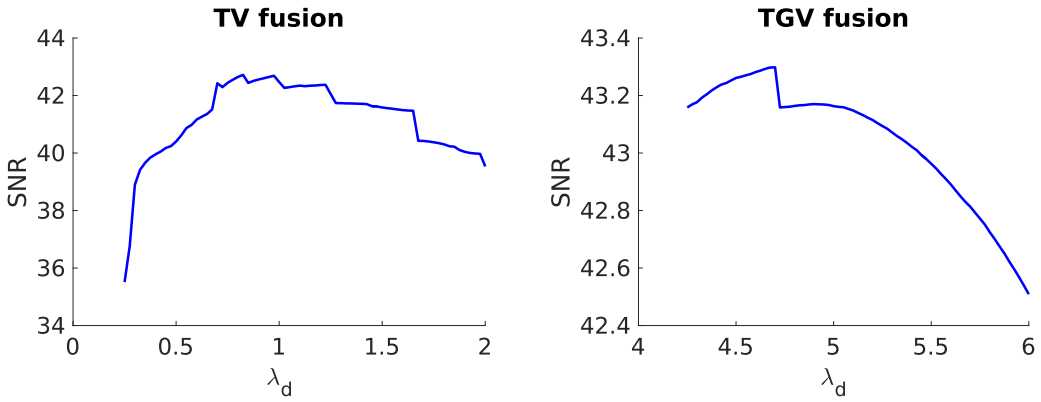
(c) Fusion using pixel based mean filtering, $SNR = 22.55$ (d) Fusion using pixel based median filtering, $SNR = 28.39$ (e) Fusion using combined medmean filtering, $SNR = 29.35$



(f) TV-L_1 fusion, $SNR = 42.72$ (g) TGV-L_1 fusion, $SNR = 43.30$

Figure 5.2: Comparison of local fusion method versus global optimization methods. Both numerical results and visual appearance show the benefit of the latter ones.

areas to zero, to simulate some uncertainty about our knowledge of these areas. As can be seen in Figure 5.4 (e) and (h), the absence of an explicit weighting results in fused DSMs with a remaining systematic error in the two modified areas, as 3 out of 5 images exhibit the same systematic offset, although with different noise. When incorporating additional prior information (here: down-weighting the image areas with the wrong offset), the optimization process is able to reconstruct the intended surface, compare Figure 5.4 (f) and (i). To obtain a fair comparison of the



(a) Groundtruth data (b) 1 of 5 noisy input images

Figure 5.3: SNR values with varying λ_d to obtain best parameter.

4 different energy functionals, we ran the algorithm for varying λ_d values and chose the parameter which resulted in the highest SNR value – compare Figure 5.4 (g) and (j). Furthermore the noise was fixed for the different runs as well.

Artificial Tests - Varying DSM resolution / Sparse DSM

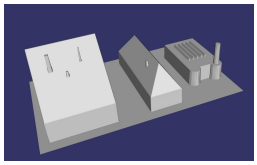
In this experiment, we compare the fusion results of the following two cases

- One noisy input DSM is given. This reduces the algorithm to a pure denoising algorithm.
- Additionally to the noisy DSM given before, an additional accurate DSM is given, exhibiting strong sparsity. This can be the result of either projecting a coarse-resolution DSM to the coordinate frame of a fine-resolution DSM or general depth priors resulting from completely different sensors as for example radar satellites.

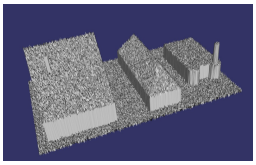
In Figure 5.5 the two abovementioned synthetic input DSMs are depicted, together with the corresponding fusion results of either using only one input DSM or adding the second sparse DSM to the optimization process as well. The latter case improves the accuracy, if not by very much. But please note that the sparsity of the second DSM is only $1/16 = 6.25\%$ compared to the first input DSM. For this experiment, both DSMs (or their valid depth pixels respectively) are weighted equally.

Unimodal DSM fusion

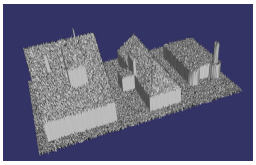
In our second evaluation, we created 14 different DSMs of the same 4.5km^2 area of the inner city of Las Vegas using a stereo reconstruction framework as proposed



(a) Groundtruth data

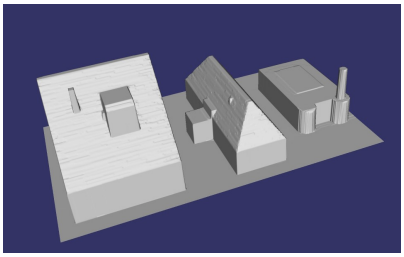


(b) 2 of 5 input images with added noise

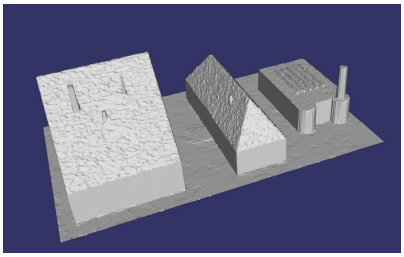


(c) 3 of 5 input images with added noise and having wrong systematic errors

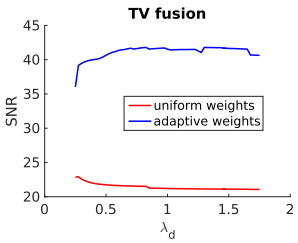
(d) Corresponding weights used in the optimization scheme: The areas with systematic error are given a weight $w = 0.2$ (dark gray), for the rest $w = 1.0$ (light gray).



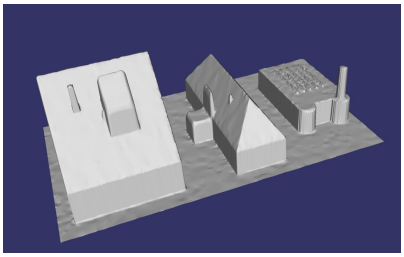
(e) TV-L_1 fusion, $SNR = 22.91$



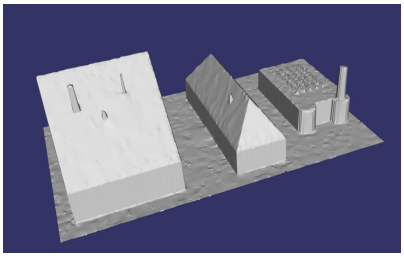
(f) Weighted TV-L_1 fusion, $SNR = 41.80$



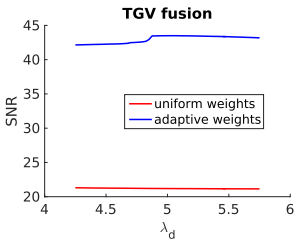
(g) Plotting SNR with varying λ_d



(h) TGV-L_1 fusion, $SNR = 21.31$



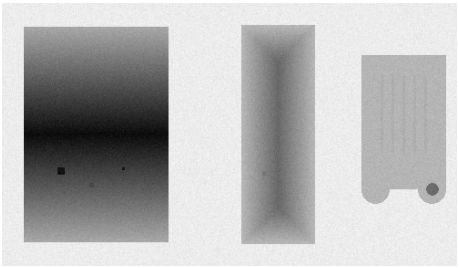
(i) Weighted TGV-L_1 fusion, $SNR = 43.50$



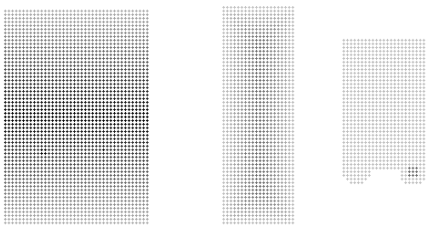
(j) Plotting SNR with varying λ_d

Figure 5.4: Evaluation of using explicit weights for simulated systematic errors in some of the input data (c). Standard TV-L_1 or TGV-L_1 fusion is not able to remove this systematic error, since it is consistent in 3 of 5 input images. When explicitly down-weighting these areas, (f) and (i), the surface is reconstructed as intended.

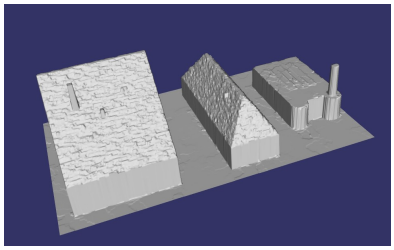
in [56]. For this we have a collection of 60 Skybox images, taken from different positions. The ground sampling distance (GSD) of these images are 1.5m and for evaluation purposes, we obtained a LiDAR measurement of the same area by aerial laser scanning having a point density of 0.375 points per m^2. As the Skybox images were taken from with a high off-nadir angle, areas behind high buildings are occluded, and cannot be reconstructed. Points in the occluded areas were not con-



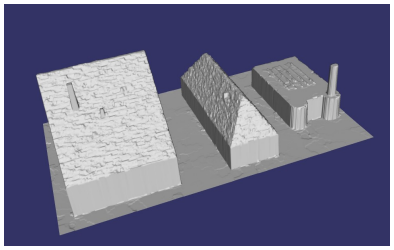
(a) Noisy input DSM #1



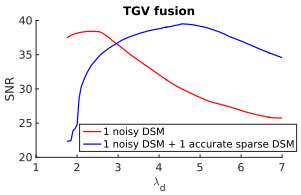
(b) Accurate but sparse input DSM #2. For illustration purpose the sampling points where thickened - the real input DSM has a sparsity of 1/16 = 6.25%.



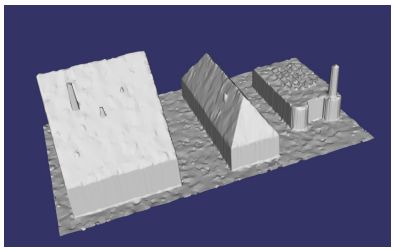
(c) TV-L_1 denoising (1 DSM), $SNR = 37.75$



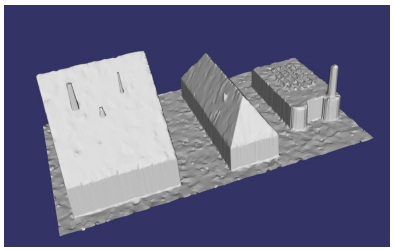
(d) TV-L_1 fusion (2 DSMs of different resolution), $SNR = 38.59$



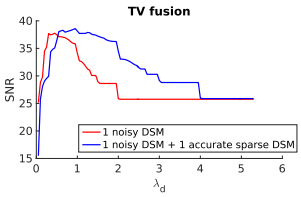
(e) Plotting SNR with varying λ_d



(f) TGV-L_1 denoising (1 DSM), $SNR = 38.41$



(g) TGV-L_1 fusion (2 DSMs of different resolution), $SNR = 39.51$



(h) Plotting SNR with varying λ_d

Figure 5.5: Evaluation of fusing DSMs with different ground sampling distance (simulated by a sparsity of 1/16 = 6.25% of the second DSM.

sidered during the statistical evaluation.

We also created 20 different DSMs of two different areas of London, using 5 in-track WorldView-2 images with a GSD of 0.5 m. First, we focused on a 1km × 1km area of the inner city of London, and second on a 1.5km × 1km park area. A LiDAR dataset, with a GSD of 1.0 m is used as reference. A satellite image of each area is shown in Figure 5.6. Figure 5.7 shows the computed fused DSMs of the inner city of London using medmean, TGV-L_1 and TV-L_1 fusion.

The accuracy of the fused DSMs with respect to the LiDAR ground truth for the Las Vegas and London data set is given in Tables 5.1, 5.2 and 5.3 in the common error metrics Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Normalized Median Absolute Deviation (NMAD). Here the improvements are hardly detectable at all, with all algorithms exhibiting similar numerical results. As of yet we do not have further explanation for these results, but strongly suspect the quality of the input DSMs, and of the LiDAR ground truth. Indeed, we noticed and removed some strong outliers in the LiDAR points, but we imagine some less strong outliers were still used during the evaluation.

	MAE [m]	RMSE [m]	NMAD [m]
Medmean	1.82	4.06	1.16
TV-L_1	1.93	4.23	1.21
TGV-L_1	1.95	4.22	1.22

Table 5.1: Las Vegas dataset: Accuracy of the fused DSM w.r.t. ground truth obtained by aerial laserscanning (LiDAR)

	MAE [m]	RMSE [m]	NMAD [m]
Medmean	1.36	2.20	1.01
TV-L_1	1.62	2.72	1.16
TGV-L_1	1.63	2.72	1.15

Table 5.2: London dataset (Inner City): Accuracy of the fused DSM w.r.t. ground truth obtained by aerial laserscanning (LiDAR)

	MAE [m]	RMSE [m]	NMAD [m]
Medmean	1.05	1.85	0.65
TV-L_1	1.13	1.99	0.67
TGV-L_1	1.17	2.06	0.68

Table 5.3: London dataset (Park): Accuracy of the fused DSM w.r.t. ground truth obtained by aerial laserscanning (LiDAR)

In fact, the statistics appear to be a little better for medmean fusion than for TGV and TV fusion. However, visual inspection of TGV and TV results show less noise and better definition of building boundaries and small streets. This may be due to the fact that for each LiDAR point, we do not calculate the z-axis distance between

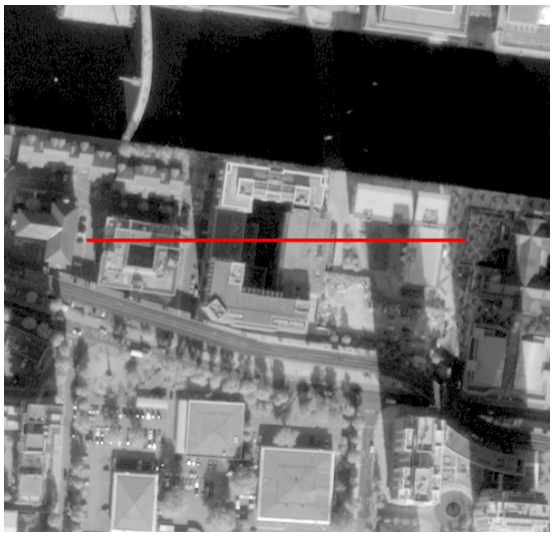
this point and the DSM, but the Euclidian distance between LiDAR point and DSM surface. This leads to not taking big outliers into account in the evaluation. For example, huge outliers located between two buildings will lead to reasonably small errors.
Furthermore, we also noticed that medmean fusion leads to a few visually erroneous results in areas for which the LiDAR data are not defined, and thus are not taken into account in the statistics. We can see those phenomena in Figure 5.8 : First, on the right side of the building (Zone A), we remark that medmean fusion yields artifacts which are not taken into account in the statistics as no LiDAR points were available for this region.
Second, on the upper edge of the building (Zone B), medmean fusion yields slowly decreasing artifacts, which are approximately 30m high, the building being 255m high, the neighbouring building 85m, and the artifacts having a height of 115m. But as we are taking the Euclidian distance into account for the evaluation, the calculated error in this place is only about 2m which is even a little smaller than for the correct TV-L_1 result.
Last, upper the building (Zone C) we notice an artifact which is a 50m high crane, and which was removed using TV-L_1. Despite this, we observe an error of about 3m for TV-L_1 fusion, and about 50cm for medmean fusion there. Moreover, we can also observe visual differences between TGV-L_1 and medmean fusion in Figure 5.9. Indeed, the edges seem to be sharper and the surfaces more regular using TGV-L_1 fusion than using medmean fusion. Finally, we also notice two points visualizing the height profiles in Figure 5.9: First, medmean fusion is indeed less smooth and contains more noise than TV-L_1 and TGV-L_1 fusion. Second, the LiDAR ground truth also contains some outlier points inside and below the buildings, which might additionally compromise the evaluation results.

Multimodal DSM fusion

Our third evaluation is investigating the results of fusing DSMs derived from different sensors and different spatial resolutions. The test data is taken from the ISPRS benchmark [91] and consists of 3 different scenes (hilly forest = *Vacarisses*, city = *Terrassa*) near Barcelona, Spain. For each scene, we compute DSMs from the a Pleiades triplet and a Worldview-1 stereo pair with a GSD of 1 m. As reference we use a LiDAR point cloud a density of 0.3 points per square meter. DSMs for all 3 possible image pairs of the Pleiades were computed and merged. To evaluate the filtering effect of TV-L_1 and TGV-L_1 the WorldView-1 DSM was additionally processed with the TV and TGV algorithms. The numerical results of local median fusion, global TV-L_1, and TGV-L_1 fusion are given in Table 5.4.

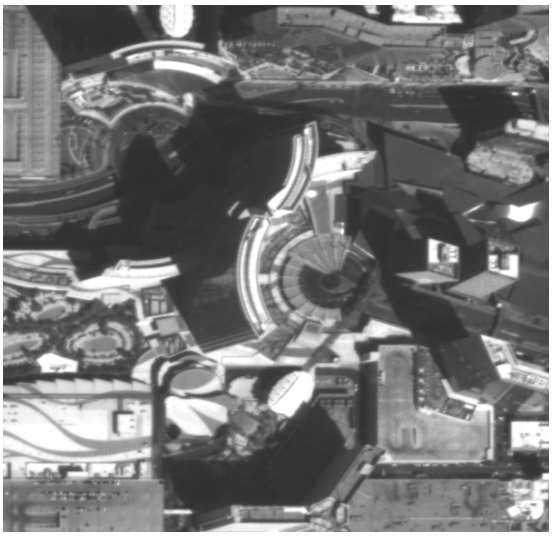
While the filtering of the WorldView-1 DSM does not significantly change the statistics for the Terrassa dataset, which to a larger extend consists of manmade structures and fields, the filter has a stronger smoothing effect on the mainly forested

(a) London inner city (1)



(b) London inner city (2)

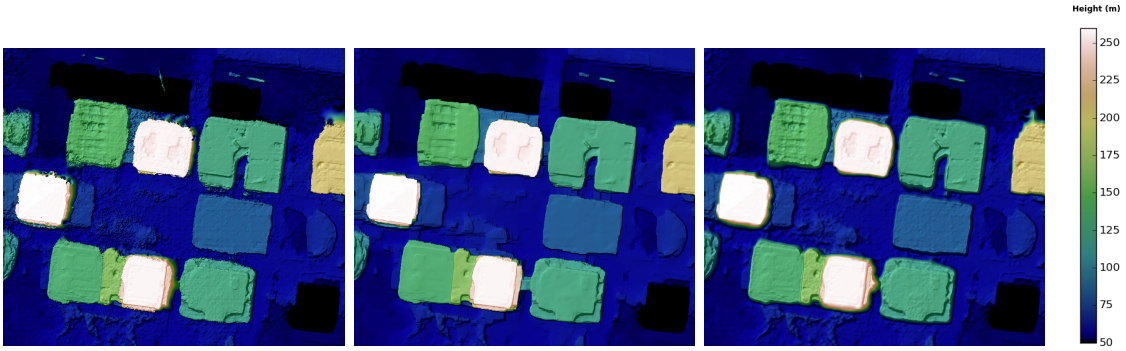
(c) London park



(d) Las Vegas

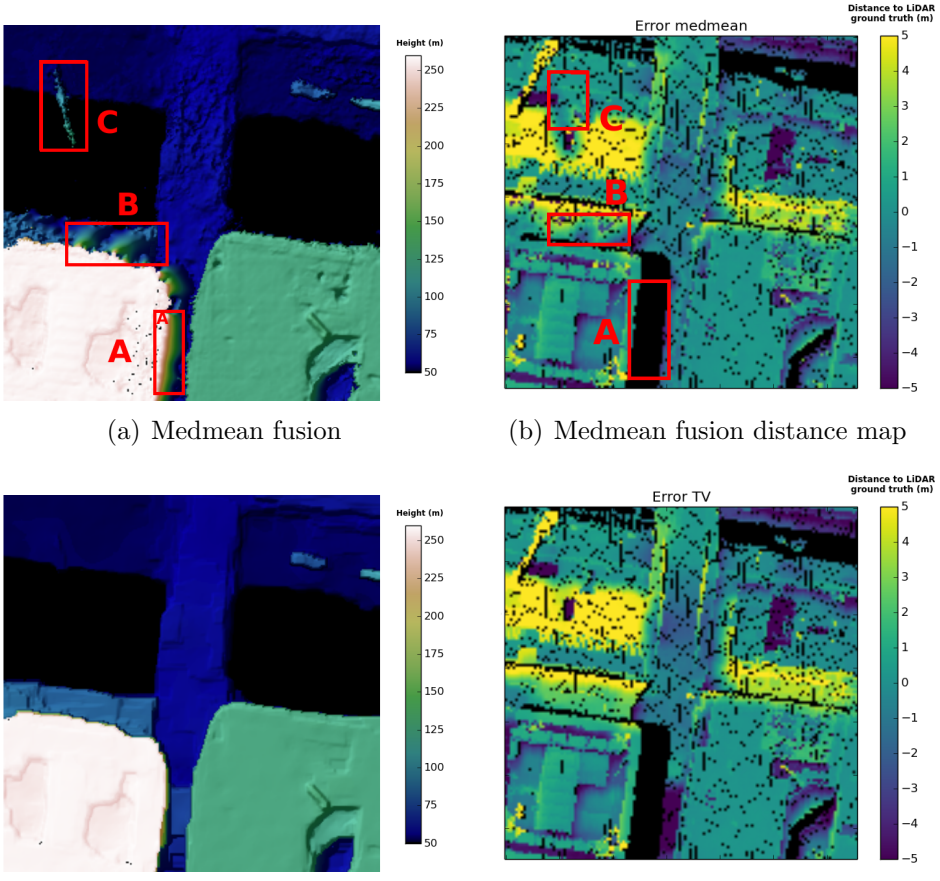
Figure 5.6: Exemplary optical images for evaluation of real-world satellite data. The red lines depict the line for the height profiles shown in Figure 5.9.

and hilly landscape of the Vacarisses area. A larger RMSE value is observed for the TGV-L_1 solution. In this special case, the TGV solution propagated outliers in the textureless shadow areas, and at steep slopes, leading to worse results. As in the London areas, objects such as building contours and bridges appear sharper, but this effect cannot be mesured properly by the relatively sparse LiDAR reference data.



(a) Medmean fusion (b) TV-L_1 fusion (c) TGV-L_1 fusion

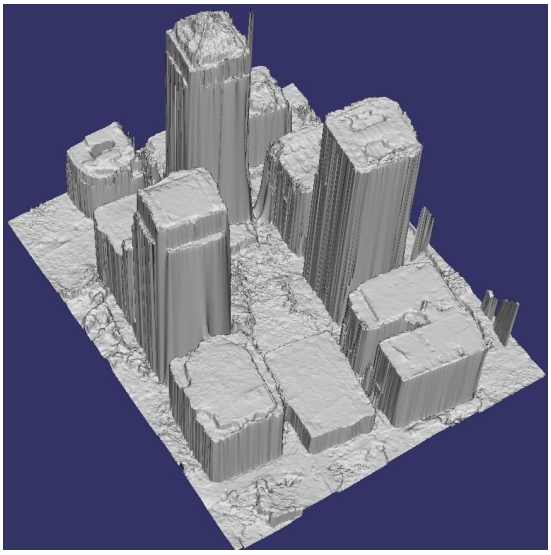
Figure 5.7: London dataset: medmean, TV-L_1 and TGV-L_1 fusion for inner city (1)



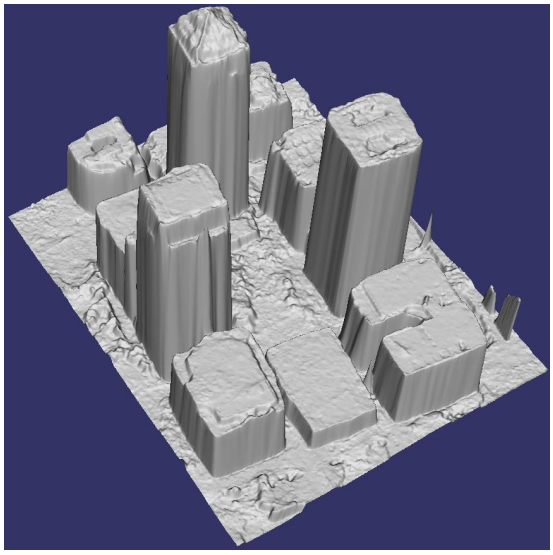
(a) Medmean fusion (b) Medmean fusion distance map

(c) TV-L_1 fusion (d) TV-L_1 fusion distance map

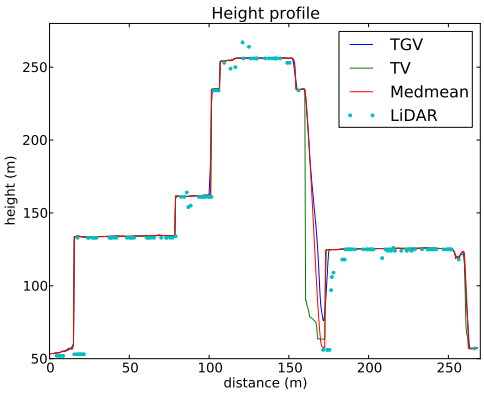
Figure 5.8: London dataset: medmean and TV-L_1 fusion together with distance to LiDAR ground truth



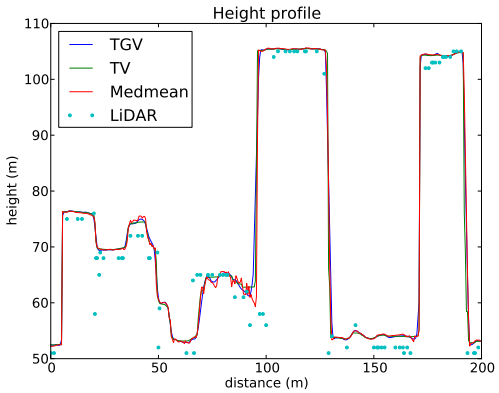
(a) Result of medmean fusion



(b) Result of TGV fusion

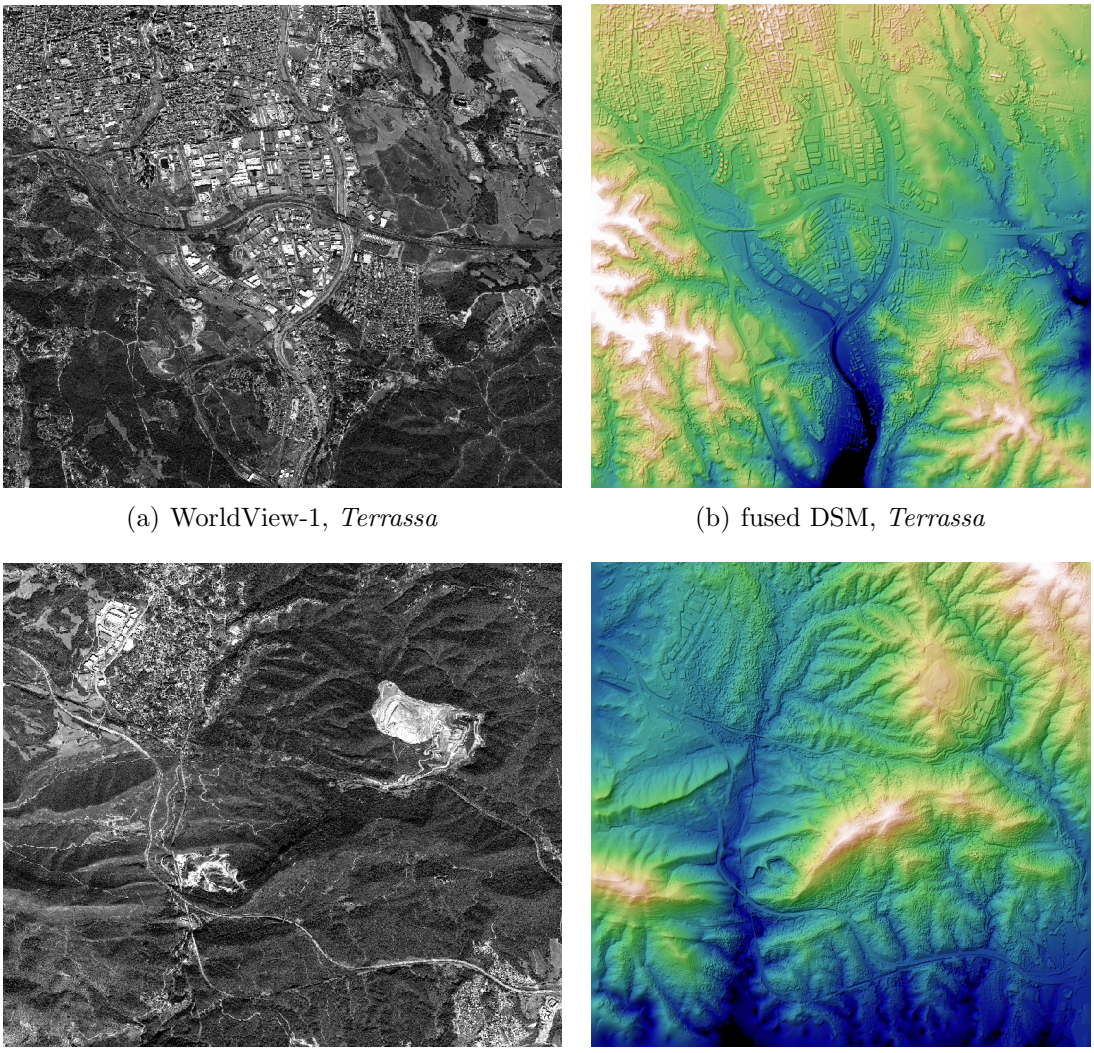


(c) Height profile, see Figure 5.6(a)



(d) Height profile, see Figure 5.6(b)

Figure 5.9: London dataset inner city: Fusion results



(a) WorldView-1, *Terrassa*

(b) fused DSM, *Terrassa*

(c) WorldView-1, *Vacarisses*

(d) fused DSM, *Vacarisses*

Figure 5.10: ISPRS dataset: Exemplary WorldView-1 images of the scenes used in the evaluation

Algorithm	Terrassa			Vacarisses		
	MAE[m]	RMSE[m]	NMAD[m]	MAE[m]	RMSE[m]	NMAD[m]
WV-1	1.05	2.23	0.59	1.62	2.88	1.09
WV-1 TV-L_1	1.04	2.20	0.59	1.81	3.48	1.11
WV-1 TGV-L_1	1.06	2.24	0.59	2.45	6.41	1.12
PL medmean	0.97	1.73	0.68	1.43	2.28	1.30
PL TV-L_1	1.03	1.84	0.67	1.58	2.54	1.38
PL TGV-L_1	1.03	1.85	0.67	1.64	2.76	1.39
PL & WV-1 medmean	0.88	1.62	0.61	1.22	1.98	1.12
PL & WV-1 TV-L_1	1.03	1.84	0.67	1.58	2.54	1.38
PL & WV-1 TGV-L_1	0.98	1.80	0.64	1.41	2.26	1.26

Table 5.4: Results of local median fusion and global TGV-L_1 fusion for heterogenous sensor data (Pleiades and WorldView-1 satellite images). The first row shows the unfused result for WorldView-1 stereo pair, the next 2 lines a "smoothing" with TV and TGV. Results for merging the individual stereo pairs of the Pleiades triplet are shown in line 3 to 6, and a fusion of Pleiades and WorldView-1 DSMs is shown in the last 3 lines.

Conclusion

In this paper we proposed a global optimization algorithms for fusing multi-resolution DSMs obtained by heterogenous sensors. These global optimization algorithms are based on adaptively weighted TV-L_1 and TGV-L_1 optimization problems, allowing for fusion of multiple DSMs enforcing additional spatial regularization. As a result, single pixels are not fused independently but a local consensus about the optimal height is achieved by taking all valid measurements in a local neighborhood into account and additionally enforcing a local planarity assumption.

In all different evaluations, both synthetic and real world data sets, a significant improvement of the visual accuracy was shown. However, the numerical accuracy is only superior for the synthetic data sets, as the ground truth for the real world data sets is too sparse and unevenly distributed - we again refer strongly to Figure 5.9 illustrating this problem. As a result, our future work will especially focus on obtaining detailed 3D ground truth within ground sampling distance of the corresponding sensors to evaluate.

Chapter 6

Conclusion

Summary

In this thesis we applied convex variational methods for depth reconstruction and fusion of depth maps, primarily using the primal-dual algorithm of [22] to solve the underlying system of differential equations. Because of its iterative and local nature the primal-dual algorithm can be parallelized to a high extent and can be solved efficiently on nowadays GPUs.

In Chapter 1 we have motivated the topic of depth reconstruction for satellite-based remote sensing applications as well as autonomous driving application. We additionally presented a review of relevant literature in the field of stereo matching cost functions, regularization techniques as well as large-scale representation of reconstructed 3D scenes. Additionally we discussed the contributions of this thesis and gave an overview of the publications within the scope of this thesis.

In Chapter 2 we highlighted the differences of the camera models between standard pinhole cameras and pushbroom cameras, gave an overview of concrete stereo matching data terms and mathematically motivated different regularization terms, resulting in energy functionals which then get numerically solved by the primal-dual algorithm.

In Chapter 3 we presented a peer-reviewed publication describing a fast algorithm for high-accuracy large-scale outdoor dense stereo reconstruction. To this end, we proposed a structure-adaptive second-order Total Generalized Variation (TGV) regularization which facilitates the emergence of planar structures by enhancing the discontinuities along building facades. As data term we used cost functions which are robust to illumination changes arising in real world scenarios. Instead of solving the arising optimization problem by a coarse-to-fine approach,

we proposed a quadratic relaxation approach which is solved by an augmented Lagrangian method. This technique allows for capturing large displacements and fine structures simultaneously. Experiments showed that the proposed augmented Lagrangian formulation leads to a speedup by about a factor of 2. The brightness-adaptive second-order regularization produces sub-disparity accurate and piecewise planar solutions, favoring not only fronto-parallel, but also slanted planes aligned with brightness edges in the resulting disparity maps. The algorithm was evaluated and shown to produce consistently good results for various data sets (close range indoor, ground based outdoor, aerial imagery).

In Chapter 4 we presented a peer-reviewed publication proposing an algorithm for dense and direct large-scale visual SLAM that runs in real-time on a commodity notebook. A fast variational dense 3D reconstruction algorithm was developed which robustly integrates data terms from multiple images. This mitigates the effect of the aperture problem and is demonstrated on synthetic and real data. An additional property of the variational reconstruction framework is the ability to integrate sparse depth priors (e.g. from RGB-D sensors or LiDAR data) into the early stages of the visual depth reconstruction, leading to an implicit sensor fusion scheme for a variable number of heterogenous depth sensors. Embedded into a keyframe-based SLAM framework, this results in a memory efficient representation of the scene and therefore (in combination with loop-closure detection and pose tracking via direct image alignment) enables us to densely reconstruct large scenes in real-time. Experimental validation on the KITTI dataset shows that our method can recover large-scale and dense reconstructions of entire street scenes in real-time from a driving car.

In Chapter 5 we presented a peer-reviewed publication proposing an algorithm for robustly fusing digital surface models (DSM) with different ground sampling distances and confidences, using explicit surface priors to obtain locally smooth surface models. Robust fusion of the DSMs is achieved by minimizing the L1-distance of each pixel of the solution to each input DSM. This approach is similar to a pixel-wise median and most outliers are discarded. We further incorporate local planarity assumption as an additional constraint to the optimization problem, thus reducing the noise compared to pixel-wise approaches. The optimization is also inherently able to include weights for the input data, therefore allowing to easily integrate invalid areas, fuse multi-resolution DSMs and to weight the input data. The complete optimization problem is constructed as a variational optimization problem with a convex energy functional, such that the solution is guaranteed to converge towards the global energy minimum. An efficient solver is presented to solve the optimization in reasonable time, e.g. running in real-time on standard computer vision camera images. The accuracy of the algorithms and the quality of the resulting

fused surface models is evaluated using synthetic datasets and spaceborne datasets from different optical satellite sensors.

Future Work

Despite actively being researched for decades, open issues in the field of depth reconstruction from optical cameras are manifold:

The most trivial task is of course improving the efficiency of existing methods, making them applicable to run on low-cost hardware and therefore enabling high accuracy depth reconstruction on-board of lightweight UAV's or self driving cars. Even if computational power in cars is increasing, only a small amount can be reserved for every one of the multiple tasks the car needs to take care of.

Another big topic is the estimation of uncertainty for depth reconstruction. Compared to depth reconstruction with active sensing via e.g. LiDAR, modelling uncertainty of the resulting depth map via optical stereo reconstruction is not straightforward and cannot be simply expressed in correlation to the distance. Although being worked on, current work such as [41], [79], [104] still relies on learning a weighted combination of heuristically chosen and manually designed uncertainty measures.

Strongly related to uncertainty is the question of auto-adaptive regularization, meaning regulating the trade-off between data term and smoothness term completely automatic and adaptively for each image area separately. Good work in this area was done by [37].

When working on image based depth reconstruction, nearly all regularizers are trying to minimize the surface based on the respective projection onto the image plane. Directly regularizing the 3D surface is physically much more correct and was being investigated by e.g. [115] and [35]. Especially the elegant latter approach unfortunately still comes with a price that regularization under perspective projection under this formulation is highly depending on the camera intrinsics and the pixel's location towards the image center.

Solving the variational problems as in this thesis still leaves a lot of room for improvement. One interesting topic would be a fast coarse-to-fine based stereo reconstruction as in chapter 4, but putting more focus on accuracy. One way of doing this would be not evaluating the image matching cost function at the current solution only (and convexifying it there), but to sample the cost function locally based on smart criteria and using a tight convex underestimator to guarantee a

good optimal solution. Work in this direction was done by [7], [6] and recently by [73], [63].

Bibliography

[1] K. Atkinson. Introduction to Modern Photogrammetry. *The Photogrammetric Record*, 18(104):329–330, 2003.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up Robust Features. *ECCV*, pages 404–417, 2006.

[3] D. Belsley. A Guide to using the Collinearity Diagnostics. *Computational Economics*, 4(1):33–50, 1991.

[4] D. Belsley, E. Kuh, and R. Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley-Interscience, 2004.

[5] D. P. Bertsekas. Constrained Optimization and Lagrange Multiplier Methods. *Computer Science and Applied Mathematics, Boston: Academic Press, 1982*, 1, 1982.

[6] A. Bhusnurmath. *Applying Convex Optimization Techniques to Energy Minimization Problems in Computer Vision*. PhD thesis, University of Pennsylvania, 2008.

[7] A. Bhusnurmath and C. J. Taylor. Solving Stereo Matching Problems using Interior Point Methods. In *Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, pages 321–329. June, 2008.

[8] S. Birchfield and C. Tomasi. A Pixel Dissimilarity Measure that is Insensitive to Image Sampling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(4):401–406, 1998.

[9] A. Blake, P. Kohli, and C. Rother. *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

[10] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *Proceedings of the British Machine Vision Conference*, pages 14–1, 2011.

[11] A. Bobick and S. Intille. Large Occlusion Stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[12] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[13] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on pattern analysis and machine intelligence*, pages 1222–1239, 2001. Graph Cuts.

[14] K. Bredies, K. Kunisch, and T. Pock. Total Generalized Variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[15] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. *Computer Vision-ECCV 2004*, pages 25–36, 2004.

[16] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. *Computer Vision - ECCV 2010*, pages 778–792, 2010.

[18] J. Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[19] A. Chambolle. Total Variation Minimization and a Class of Binary MRF Models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 136–152. Springer, 2005.

[20] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An Introduction to Total Variation for Image Analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.

[21] A. Chambolle, D. Cremers, and T. Pock. *A Convex Approach for Computing Minimal Partitions*. 2008.

[22] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, pages 1–26, 2011.

[23] R. T. Collins. A space-sweep approach to true multi-image matching. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR 1996, 1996 IEEE Computer Society Conference on*, pages 358–363. IEEE, 1996.

[24] P. d'Angelo and G. Kuschk. Dense Multi-View Stereo from Satellite Imagery. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 6944–6947. IEEE, 2012.

[25] P. d'Angelo, G. Kuschk, and P. Reinartz. Evaluation of Skybox Video and Still Image products. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:95–99, 2014.

[26] K. Davydova, G. Kuschk, H. L, R. P, and S. U. Consistent Multi-View Texturing of Detailed 3D Surface Models. *ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences, pp. 25-31*, 2015.

[27] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[28] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[29] J. Engel, J. Stueckler, and D. Cremers. Large-Scale Direct SLAM with Stereo Cameras. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1935–1942. IEEE, 2015.

[30] D. Ferstl, R. Ranftl, M. Rüther, and H. Bischof. Multi-modality depth map fusion using primal-dual optimization. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[31] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014.

[32] B. Froba and A. Ernst. Face Detection with the Modified Census Transform. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 91–96. IEEE, 2004.

[33] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[34] A. Geiger, P. Lenz, and R. Urtasun. Are we Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[35] G. Graber, J. Balzer, S. Soatto, and T. Pock. Efficient Minimal-Surface Regularization of Perspective Depth Maps in Variational Stereo. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn*, volume 4, 2015.

[36] G. Graber, T. Pock, and H. Bischof. Online 3D Reconstruction using Convex Optimization. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 708–711. IEEE, 2011.

[37] M. Grasmair. Locally Adaptive Total Variation Regularization. In *Scale Space and Variational methods in computer Vision*, pages 331–342. Springer, 2009.

[38] J. Grodecki. Ikonos Stereo Feature Extraction - RPC Approach. In *Proc. ASPRS Annual Conference, St. Louis*, pages 23–27, 2001.

[39] J. Grodecki and G. Dial. IKONOS Geometric Accuracy. In *Proceedings of Joint Workshop of ISPRS Working Groups I/2, I/5 and IV/7 on High Resolution Mapping from Space*, pages 19–21, 2001.

[40] J. Grodecki and G. Dial. Block Adjustment of High-Resolution Satellite Images Described by Rational Functions. *Photogrammetric Engineering and Remote Sensing*, 69(69):59–70, 2003.

[41] R. Haeusler, R. Nair, and D. Kondermann. Ensemble Learning for Confidence Measures in Stereo Vision. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 305–312. IEEE, 2013.

[42] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[43] P. Heise, B. Jensen, S. Klose, and A. Knoll. Variational PatchMatch MultiView Reconstruction and Refinement. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 882–890, 2015.

[44] H. Hirschmueller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005.

[45] H. Hirschmueller. Stereo Vision in Structured Environments by Consistent Semi-Global Matching. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2386–2393. IEEE, 2006. semi global matching.

[46] H. Hirschmueller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 328–341, 2008.

[47] H. Hirschmueller and S. Gehrig. Stereo Matching in the Presence of Sub-Pixel Calibration Errors. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 437–444. IEEE, 2009.

[48] H. Hirschmueller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. In *Computer Vision and Pattern Recognition, 2007. CVPR 2007. IEEE Conference on*, pages 1–8. IEEE, 2007.

[49] H. Hirschmueller and D. Scharstein. Evaluation of Stereo Matching Costs on Images with Radiometric Differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1582–1599, 2009.

[50] H. Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1333–1336, 2003.

[51] J. Kim, V. Kolmogorov, and R. Zabih. Visual Correspondence using Energy Minimization and Mutual Information. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1033–1040. IEEE, 2003.

[52] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. Turning Mobile Phones into 3D Scanners. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3946–3953. IEEE, 2014.

[53] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004.

[54] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gussefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, et al. The HCI Benchmark Suite: Stereo And Flow Ground Truth With Uncertainties for Urban Autonomous Driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–28, 2016.

[55] T. Krauss, P. d'Angelo, K. G, T. J, and P. T. 3D-Information Fusion from Very High Resolution Satellite Sensors. *Proceedings of International Symposium on Remote Sensing of Environment (ISRSE) 2015*, 2015.

[56] G. Kuschk. Large Scale Urban Reconstruction from Remote Sensing Imagery. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5:W1, 2013.

[57] G. Kuschk. Model-Free Dense Stereo Reconstruction for Creating Realistic 3D City Models. In *Urban Remote Sensing Event (JURSE), 2013 Joint*, pages 202–205. IEEE, 2013.

[58] G. Kuschk, A. Božič, and D. Cremers. Real-time Variational Stereo Reconstruction with Applications to Large-scale Dense SLAM. pages 1348–1355, 2017.

[59] G. Kuschk and D. Cremers. Fast and Accurate Large-scale Stereo Reconstruction using Variational Methods. In *ICCV Workshop on Big Data in 3D Computer Vision*, Sydney, Australia, December 2013.

[60] G. Kuschk and P. d'Angelo. Fusion of Multi-Resolution Digital Surface Models. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W3:247–251, 2013.

[61] G. Kuschk, P. d'Angelo, D. Gaudrie, P. Reinartz, and D. Cremers. Spatially Regularized Fusion of Multiresolution Digital Surface Models. *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[62] G. Kuschk, P. d'Angelo, R. Qin, D. Polic, P. Reinartz, and D. Cremers. DSM Accuracy Evaluation for the ISPRS Commission I Image Matching Benchmark. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:195–200, 2014.

[63] E. Laude, T. Moellenhoff, M. Moeller, J. Lellmann, and D. Cremers. Sublabel-accurate convex relaxation of vectorial multilabel energies. In *Computer Vision–ECCV 2016*, October 2016.

[64] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[65] Y. Li and S. Osher. A New Median Formula with Applications to PDE Based Denoising. *Commun. Math. Sci*, 7(3):741–753, 2009.

[66] C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer SocietyConference on.*, volume 1. IEEE, 2002.

[67] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[68] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[69] C. McGlone, E. Mikhail, and J. Bethel. Manual of Photogrammetry. American Society for Photogrammetry and Remote Sensing, 1980.

[70] M. Meilland and A. I. Comport. On Unifying Key-frame and Voxel-based Dense Visual SLAM at Large Scales. In *IEEE Int. Conf. on Intelligent Robots and Systems*, volume 2, page 5, 2013.

[71] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

[72] O. Meynberg and G. Kuschk. Airborne Crowd Density Estimation. In *CMRT13 - City Models, Roads and Traffic 2013*, volume II-3/W3 of *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 49–54. Copernicus GmbH, November 2013.

[73] T. Moellenhoff, E. Laude, M. Moeller, J. Lellmann, and D. Cremers. Sublabel-accurate relaxation of nonconvex energies. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016.

[74] H.-H. Nagel and W. Enkelmann. An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5):565–593, 1986.

[75] R. Newcombe, S. Lovegrove, and A. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[76] M. Niessner, M. Zollhoefer, S. Izadi, and M. Stamminger. Real-time 3D Reconstruction at Scale using Voxel Hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013.

[77] J. Oh. *Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-Based Georeferencing of Aerial Images*. PhD thesis, The Ohio State University, 2011.

[78] H. Papasaika, E. Kokiopoulou, E. Baltsavias, K. Schindler, and D. Kressner. Fusion of Digital Elevation Models using Sparse Representations. *Photogrammetric Image Analysis*, pages 171–184, 2011.

[79] M.-G. Park and K.-J. Yoon. Leveraging Stereo Matching with Learning-based Confidence Measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 101–109, 2015.

[80] R. Perko and C. Zach. Globally optimal robust dsm fusion. *European Journal of Remote Sensing*, 49:489–511, 2016.

[81] M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2609–2616. IEEE, 2014.

[82] T. Pock and A. Chambolle. Diagonal Preconditioning for First Order Primal-Dual Algorithms in Convex Optimization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1762–1769. IEEE, 2011.

[83] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A Convex Relaxation Approach for Computing Minimal Partitions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 810–817. IEEE, 2009.

[84] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An Algorithm for Minimizing the Mumford-Shah Functional. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1133–1140. IEEE, 2009.

[85] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global Solutions of Variational Models with Convex Regularization. *SIAM Journal on Imaging Sciences*, 3(4):1122–1145, 2010.

[86] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A Convex Formulation of Continuous Multi-Label Problems. *Computer Vision - ECCV 2008*, pages 792–805, 2008.

[87] T. Pock, L. Zebedin, and H. Bischof. TGV-Fusion. *Rainbow of computer science*, pages 245–258, 2011.

[88] R. Ranftl, K. Bredies, and T. Pock. Non-Local Total Generalized Variation for Optical Flow Estimation. In *Computer Vision–ECCV 2014*, pages 439–454. Springer, 2014.

[89] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the Limits of Stereo using Variational Stereo Estimation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 401–407. IEEE, 2012.

[90] R. Ranftl, T. Pock, and H. Bischof. Minimizing TGV-Based Variational Models with Non-convex Data Terms. In *Scale Space and Variational Methods in Computer Vision*, pages 282–293. Springer, 2013.

[91] P. Reinartz, P. d'Angelo, T. Krauß, D. Poli, K. Jacobsen, and G. Buyuksalih. Benchmarking and Quality Analysis of DEM Generated from High and Very High Resolution Optical Stereo Satellite Data. In *ISPRS Symposium Commission I*, 2010.

[92] P. Reinartz, R. Mueller, D. Hoja, M. Lehner, and M. Schroeder. Comparison and Fusion of DEM Derived from SPOT-5 HRS and SRTM Data and Estimation of Forest Heights. In *Proc. EARSeL Workshop on 3D-Remote Sensing, Porto*, 2005.

[93] P. Reinartz, J. Tian, H. Arefi, T. Krauss, G. Kuschk, T. Partovi, and P. d'Angelo. Advances in DSM Generation and Higher Level Information Extraction from High Resolution Optical Stereo Satellite Data. *34th Earsel Symposium*, 34:1–9, 2014.

[94] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *International Conference on Computer Vision (ICCV)*, 2017.

[95] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016.

[96] L. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[97] M. Rumpler, A. Irschara, A. Wendel, and H. Bischof. Rapid 3d city model approximation from publicly available geographic data sources and georeferenced aerial images. In *Computer vision winter workshop (CVWW)*, 2012.

[98] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth]. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.

[99] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International journal of computer vision*, 47(1):7–42, 2002.

[100] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

[101] K. Schindler, H. Papasaika-Hanusch, S. Schuetz, and E. Baltsavias. Improving Wide-Area DEMs Through Data Fusion - Chances and Limits. *Proceedings of the 54th Photogrammetric Week, Stuttgart, Germany*, 2011.

[102] T. Schoeps, T. Sattler, C. Haeane, and M. Pollefeys. 3D Modeling on the Go: Interactive 3D Reconstruction of Large-Scale Scenes on Mobile Devices. In *3D Vision (3DV), 2015 International Conference on*, pages 291–299. IEEE, 2015.

[103] H. Schultz, E. M. Riseman, F. R. Stolle, and D.-M. Woo. Error Detection and DEM Fusion using Self-Consistency. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE InternationalConference on*, volume 2, pages 1174–1181. IEEE, 1999.

[104] A. Spyropoulos, N. Komodakis, and P. Mordohai. Learning to Detect Ground Control Points for Improving the Accuracy of Stereo Matching. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1621–1628. IEEE, 2014.

[105] F. Stein. Efficient Computation of Optical Flow using the Census Transform. In *Joint Pattern Recognition Symposium*, pages 79–86. Springer, 2004.

[106] F. Steinbruecker, C. Kerl, J. Sturm, and D. Cremers. Large-Scale Multi-Resolution Surface Reconstruction from RGB-D Sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3264–3271, 2013.

[107] F. Steinbruecker, T. Pock, and D. Cremers. Large Displacement Optical Flow Computation without Warping. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1609–1614. IEEE, 2009.

[108] C. Straub, J. Tian, R. Seitz, and P. Reinartz. Assessment of cartosat-1 and worldview-2 stereo imagery in combination with a lidar-dtm for timber volume estimation in a highly structured forest in germany. *Forestry*, 86(4):463–473, 2013.

[109] J. Stueckler and S. Behnke. Multi-Resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147, 2014.

[110] J. Stuehmer, S. Gumhold, and D. Cremers. Real-Time Dense Geometry from a Handheld Camera. *Joint Pattern Recognition Symposium*, pages 11–20, 2010.

[111] J. Sun, N. Zheng, and H. Shum. Stereo Matching using Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 787–800, 2003. Belief propagation.

[112] H. Sunyoto, W. Van der Mark, and D. Gavrila. A Comparative Study of Fast Dense Stereo Vision Algorithms. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 319–324. Ieee, 2004.

[113] A. N. Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

[114] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *IEEE transactions on pattern analysis and machine intelligence*, pages 815–830, 2009.

[115] B. Ummenhofer and T. Brox. Global, Dense Multiscale Reconstruction for a Billion Points. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1341–1349, 2015.

[116] P. Viola and W. Wells III. Alignment by Maximization of Mutual Information. *International journal of computer vision*, 24(2):137–154, 1997.

[117] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[118] Q. Wang, A. Gaidon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[119] M. Werlberger, T. Pock, and H. Bischof. Motion Estimation with Non-Local Total Variation Regularization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2464–2471. IEEE, 2010.

[120] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 Optical Flow. In *Proceedings of the British machine vision conference*, volume 34, pages 1–11. Citeseer, 2009.

[121] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM without a Pose Graph. In *Proceedings of Robotics: Science and Systems (RSS)*, 2015.

[122] Q. Yang, L. Wang, and N. Ahuja. A Constant-Space Belief Propagation Algorithm for Stereo Matching. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conferenceon*, pages 1458–1465. IEEE, 2010.

[123] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister. Real-Time Global Stereo Matching using Hierarchical Belief Propagation. In *The British Machine Vision Conference*, pages 989–998, 2006.

[124] K. Yoon and I. Kweon. Adaptive Support-weight Approach for Correspondence Search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):650–656, 2006.

[125] R. Zabih and J. Woodfill. Non-parametric Local Transforms for Computing Visual Correspondence. *Computer Vision - ECCV 1994*, pages 151–158, 1994.

[126] C. Zach. Fast and High Quality Fusion of Depth Maps. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, volume 1, 2008.

[127] C. Zach, T. Pock, and H. Bischof. A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conferenceon*, pages 1–8. IEEE, 2007.

[128] Z. Zhang. Determining the Epipolar Geometry and its Uncertainty: A Review. *International Journal of Computer Vision*, 27(2):161–195, 1998.

[129] Z. Zhang. A Flexible New Technique for Camera Calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2002.