



Technische Universität München  
DEPARTMENT OF MATHEMATICS

# Multi-View Stereo Reconstruction

## Variational Fusion of Depth Maps

Master's Thesis  
by  
Christina Eisenhofer

Supervisor: Prof. Dr. Oliver Junge  
Advisor: Prof. Dr. Carsten Steger  
Submission date: June 7, 2013





Ich erkläre hiermit, dass ich die Master's Thesis selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe.

München, den 07. Juni 2013

---

Christina Eisenhofer



## Abstract

Multi-view stereo aims to reconstruct a complete 3D object model from a set of images acquired from known camera positions. We present different classes of reconstruction algorithms and motivate the choice of one particular variational approach. This method merges multiple depth maps and is based on minimizing an energy functional that combines the robust  $L^1$ -norm with total variation regularization. We analyze the energy model, present an optimization scheme, and prove the global convergence to an optimal solution. The evaluation on several multi-view stereo data sets shows that high quality 3D models can be obtained within few minutes. Finally, possible methods to increase the performance of the reconstruction process are addressed.

## Zusammenfassung

Die vorliegende Arbeit befasst sich mit der Mehrbild-Stereo Rekonstruktion eines Objekts. Hierbei wird mit Hilfe von Bildern aus verschiedenen Kamerapositionen ein vollständiges 3D-Objektmodell erstellt. Wir geben einen Überblick über unterschiedliche Mehrbild-Stereo Verfahren und begründen die Wahl eines Variationsansatzes, der auf Tiefenkarten basiert. Dieser minimiert ein Kostenfunktional, das aus einem  $L^1$ - Datenterm und der Totalvariation als Strafterm besteht. Wir analysieren dieses Modell und stellen einen Algorithmus zur Lösung des Minimierungsproblems vor. Es kann gezeigt werden, dass dieser konvergiert und eine globale Lösung liefert. Das Verfahren wird für verschiedene Datensätze getestet und evaluiert. Abschließend werden Möglichkeiten zur Laufzeitverbesserung diskutiert.



---

## Acknowledgment

---

This thesis was conducted at MVTec Software GmbH in cooperation with Technische Universität München. First and foremost, I want to express my sincere gratitude to my supervisors Prof. Dr. Oliver Junge and Prof. Dr. Carsten Steger, who made this collaboration possible and were supporting me at any time.

I want to thank Prof. Dr. Carsten Steger for providing an excellent working environment. I am very grateful for his guidance, support, and detailed suggestions of improvement.

Furthermore, I want to thank my colleagues and friends Tobias Böttger and Matthias Meßner for their valuable help. I am very thankful for many fruitful discussions and for spending a great time together.

My special thanks go to my sister, who took the time to give me numerous professional advices as well as personal encouragement.

I also want to thank Jakob Zimmermann, Andreas Pumpf, Andre Milzarek, and Roman Moie for giving me helpful advices and for answering all my questions patiently.

Finally, I would like to thank my family and friends for their assistance and support while working on this thesis.





<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	3D Object Models . . . . .	1
1.2	Requirements and Goals . . . . .	3
1.3	Multi-View Stereo Reconstruction . . . . .	3
1.3.1	Middlebury Evaluation . . . . .	4
1.3.2	Reconstruction Approachs . . . . .	5
1.4	Selected Method . . . . .	13
<b>2</b>	<b>Convex Functionals and Minimization Problems</b>	<b>15</b>
2.1	Extended Real-Valued Functionals . . . . .	15
2.2	Convex Functionals . . . . .	16
2.2.1	The Subdifferential . . . . .	17
2.2.2	The Fenchel Conjugate . . . . .	18
<b>3</b>	<b>Total Variation in Image Processing</b>	<b>21</b>
3.1	Variational Methods in Image Processing . . . . .	21
3.1.1	A Continuous Image Model . . . . .	22
3.1.2	Variational Problems . . . . .	22
3.1.3	Variational Denoising . . . . .	22
3.2	Total Variation in the Continuous Setting . . . . .	23
3.3	Total Variation in the Discrete Setting . . . . .	26
3.3.1	Discrete Image Model . . . . .	26
3.3.2	Discretization of the Total Variation . . . . .	27
3.4	An Algorithm to Solve the Variational Denoising Problem . . . . .	28
3.4.1	Derivation of the Dual Formulation . . . . .	28
3.4.2	The Dual Problem . . . . .	29
3.4.3	Discrete Formulation and Optimization . . . . .	30
3.5	Further Applications of Total Variation Regularization . . . . .	33
<b>4</b>	<b>3D Reconstruction by a Robust Variational Merging of Depth Maps</b>	<b>35</b>
4.1	TV- $L^1$ Energy Model . . . . .	35
4.2	Conversion of Depth Maps into Distance Fields . . . . .	36
4.3	Optimization Scheme . . . . .	37

4.3.1	Discrete Optimization Problem . . . . .	37
4.3.2	Approximation of the Energy Functional . . . . .	38
4.3.3	Alternating Descent Approach . . . . .	40
4.3.4	Denoising Step . . . . .	43
4.3.5	Generalized Thresholding Step . . . . .	43
4.4	Reconstruction Algorithm . . . . .	44
<b>5</b>	<b>Numerical Analysis</b>	<b>47</b>
5.1	Implementation Details . . . . .	47
5.1.1	Data Compression . . . . .	47
5.1.2	Efficient Implementation of the Generalized Thresholding Step . . . . .	48
5.2	Experimental Validation . . . . .	50
5.2.1	Data Sets . . . . .	52
5.2.2	Depth Map Estimation . . . . .	52
5.2.3	Results . . . . .	53
<b>6</b>	<b>Conclusion and Outlook</b>	<b>61</b>
6.1	GPU-friendly Approximation of the Energy Functional . . . . .	61
6.2	Variational Denoising Revisited . . . . .	62
6.3	Additional Uses of the TV- $L^1$ -Energy Model . . . . .	63

The reconstruction of 3D objects is a broad research area in computer vision that has made remarkable progress during the last years. Due to innovations in computer graphics and technology, 3D shape models are increasingly in demand. They are used in various fields like entertainment (e.g., television, film, video-games), medicine (e.g., 3D representation of organs, surgery planning/assistance), science and engineering (e.g., illustration and simulation of constructions, machines, chemical compounds, physical processes), robotics (e.g., navigation, object recognition), art and heritage (e.g., demonstration of cultural artifacts in museums), industrial design (e.g., 3D design, prototyping) and architecture (e.g., presentation of buildings).

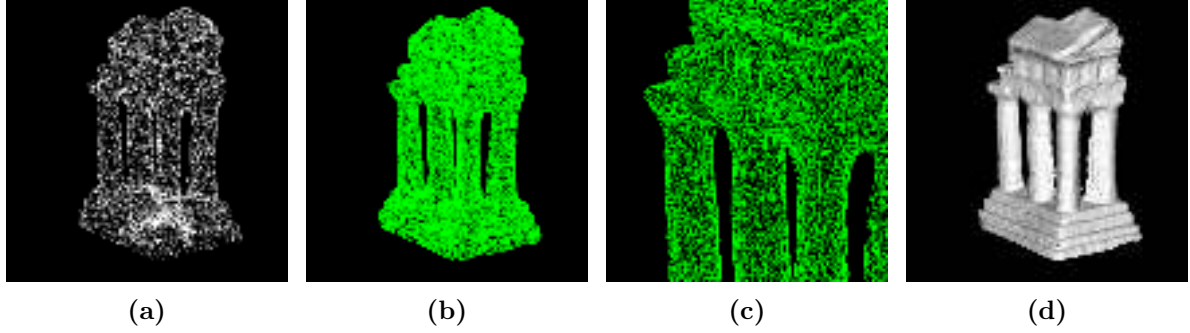
For years, high quality 3D object models could only be obtained by using active 3D acquisition methods like laser scanning or structured light techniques. Recently, however, passive methods that use multiple images acquired from different camera positions around an object have also succeeded in producing satisfying results. This so-called *multi-view stereo reconstruction* merges the image information received from different viewpoints to create an appropriate 3D shape. There are many different approaches to obtain depth information from 2D images that rely on different mathematical representations of 3D surfaces. The aim of this work is to present, analyze, and implement an accurate and still efficient reconstruction method.

In this chapter we introduce 3D object models, demonstrate the advantages of multi-view stereo in general, and present different classes of reconstruction algorithms. Furthermore, we define requirements that should be met by an appropriate method. Finally, we motivate the restriction to a special group of algorithms and, more precisely, the choice of one particular approach.

## 1.1 3D Object Models

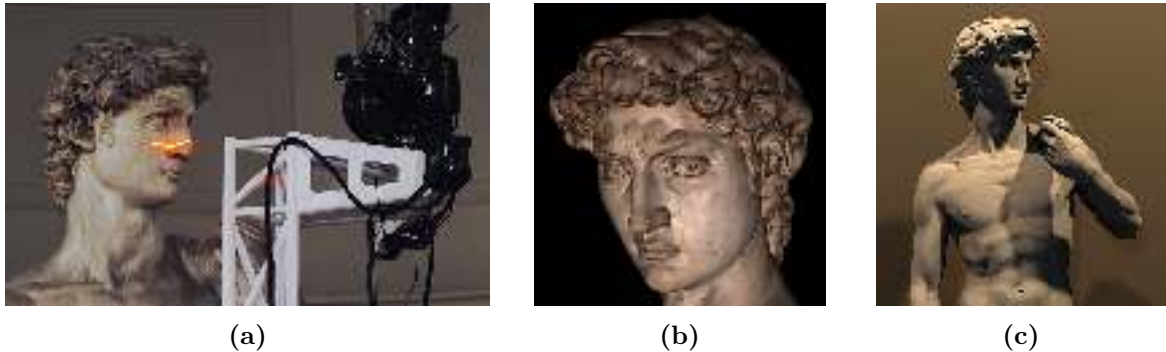
There are several ways of modeling 3D objects. For some applications, like medical imaging, complete 3D volumes are required. In these cases, information about the interior of an object must be provided and a solid 3D model must be computed. In many situations, however, reconstructing only the object surface is sufficient. Using ordinary digital images, we restrict ourselves to the latter class of problems.

In a mathematical framework, surfaces can be modeled by point clouds, level sets, voxel grids,



**Figure 1.1:** Examples of 3D model representations. (a) Point cloud (18750 points). (b) Triangulation of the points in (a) (23502 triangles). (c) Close up of the triangles. (d) Triangle mesh with colored triangle surfaces (518738 triangles).

depth images, or polygon meshes, to name only some representations. In computer graphics, usually triangle meshes or, more generally, polygon meshes are used to display 3D objects in a visually appealing way. Therefore, the outputs of most 3D reconstruction algorithms are appropriate polygon meshes. See Fig. 1.1 for a representation of an object by a point cloud (Fig. 1.1a) and the corresponding triangle mesh (Fig. 1.1b). The close up in Fig. 1.1c provides a more detailed view of the individual triangles. Note that the more points exist the more regular the triangle mesh is, yielding a smoother surface, as demonstrated in Fig. 1.1d, where the opaque triangle surfaces are colored. The obtained 3D model can further be equipped with color and texture information by performing texture mapping (see, e.g., [27]).



**Figure 1.2:** The reconstruction of Michelangelo’s David realized by a research group from Stanford University. (a) Setup of the scanning devices. (b) and (c) Reconstruction result: a colored 3D model consisting of two billion polygons. The images are obtained from the webpage of the Michelangelo project [36] by permission of Marc Levoy.

In Fig. 1.2b and c we present an impressive (colored) 3D model of Michelangelo’s “David” that consists of two billion polygons. It has been created in the context of the “Digital Michelangelo Project” [36], [37] in 1999. A research group from Stanford University used laser scans to reconstruct some of Michelangelo’s statues in Florence. Their results present excellent examples of the high quality of laser-based 3D reconstructions. Usually laser scanning is, however, time consuming. The whole scan of the David statue took 30 nights and was very complex owing to size and location of the statue, see Fig. 1.2a. In general, laser scans are often restricted to indoor scenes and limited in spatial and temporal resolution. Moreover, they do not contain any color

information. To get a colored reconstruction like in Fig. 1.2 additional color images of the object are necessary.

An alternative to laser scanning is the reconstruction of 3D shapes using images of an object taken from different points of view. This can also be done for big objects, objects that cannot be reached directly, or outdoor scenes. Moreover, digital cameras are much cheaper than laser scanners. The high demand on the consumer market leads to higher resolutions for lower prices. Even the cameras included in mobile phones are of increasing quality. These advantages motivate the usage of digital images for 3D reconstruction. While the corresponding results could not compete with 3D models obtained from laser scans for years, recently many improvements in accuracy and speed have been made. Thus, using images for 3D modeling becomes more and more attractive.

## 1.2 Requirements and Goals

This work has been developed in cooperation with MVTec Software GmbH<sup>1</sup>, which is specialized in producing software products for machine vision. Therefore, we will concentrate on the usage of 3D models in machine vision systems. Here, possible applications are object recognition and inspection. They can be used, for example, in manufacturing where a robot is guided to grasp and assemble certain 3D objects. In contrast to image-based processes, systems that use 3D models are able to determine 3D features like the object's volume or other three-dimensional moments. Consequently, the whole shape information can be obtained and verified and erroneous products can be rejected.

For industrial applications, 3D models must be obtained as quickly as possible and still be sufficiently accurate. This yields the typical tradeoff between quality and computation time. We examined several 3D reconstruction methods with particular focus on their runtimes. In what follows, we will present different conceptual types of surface reconstruction approaches and highlight their advantages and disadvantages with respect to applications in machine vision systems.

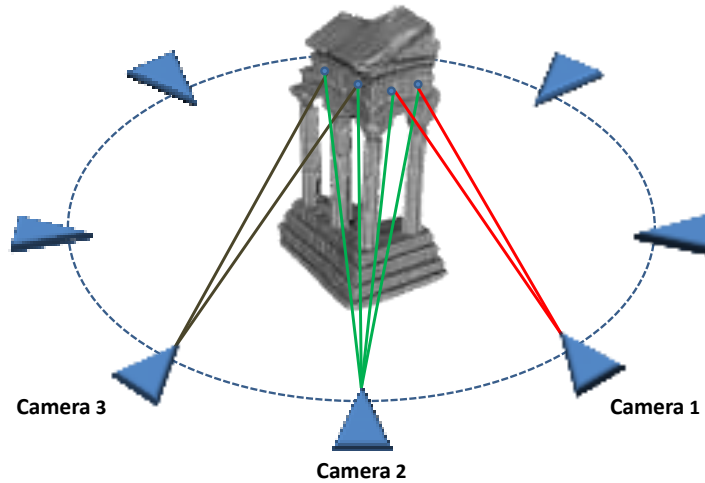
## 1.3 Multi-View Stereo Reconstruction

We will concentrate on methods that make use of images taken from known camera positions, so-called *multi-view stereo reconstruction* techniques. Hence, we assume that the used cameras are calibrated, that is, the intrinsic camera parameters and the camera poses are given. For a detailed introduction to camera models and calibration, we refer to [26, Chapter 6]. It is also possible to reconstruct objects using images from uncalibrated cameras. This leads to a different problem that is known as *structure from motion* and we recommend [26, Chapter 18-19] for further information.

As indicated in Fig. 1.3, the multi-view stereo reconstruction problem is the inversion of the image acquisition process. Since all 3D points that lie on one line of sight are projected onto the same image point, this problem is ill-posed. Therefore, reconstruction methods must perform some kind of regularization to obtain a solution. Starting from simple multiple-baseline stereo algorithms [16], [43] and discrete voxel based reconstructions like voxel coloring [49] and space carving [33], over the last decade more complex approaches solving global optimization problems have been developed.

---

<sup>1</sup>MVTec Software GmbH, Neherstr.1, 81675 München, [www.mvtec.com](http://www.mvtec.com)



**Figure 1.3:** The concept of multi-view stereo reconstruction. Multiple images are acquired from different known camera positions. Making use of the fact that in general a 3D point is visible in several images, 3D information can be extracted and the surface of an object can be reconstructed.

### 1.3.1 Middlebury Evaluation

In 2006, Seitz et al. [48] (Middlebury College, Vermont, USA) provided an evaluation of multi-view stereo reconstruction algorithms, which for the first time made it possible to quantitatively compare different methods. This has evoked a big improvement in accuracy and performance during the last years. The corresponding so-called *Middlebury data sets* together with evaluation results are publicly available on [46] and currently updated. The “Temple” and “Dino” data sets consist of a set of images (temple: 16, 47 and 312 views, dino: 16, 48 and 363 views), the camera parameters and a 3D ground truth model captured by a laser stripe scanner, respectively. Some of the images and the corresponding ground truth models are shown in Fig. 1.4. Seitz et al. introduced methods to compute both the accuracy and the completeness of the submitted reconstructions. Here, accuracy describes how close a solution  $S$  is to the ground truth model  $G$ . It is computed as the distance  $d$  such that a fixed percentage  $X$  (e.g.  $X = 90\%$ ) of points of  $S$  is contained within distance  $d$  of the ground truth  $G$ . Furthermore, completeness reflects how much of  $G$  is contained in  $S$  and is measured as the percent of points of  $G$  that are within a fixed distance  $x$  (e.g.  $x = 1.25$  mm) of the reconstruction  $S$ . For almost all state-of-the-art multi-view stereo methods, these values have been analyzed and, together with the reconstruction results and runtimes, they can not only be found in the corresponding publications but also on the Middlebury webpage [46]. This makes it possible to compare different techniques and to draw conclusions about certain classes of reconstruction algorithms, as will be done in Section 1.3.2. Since it is the standard benchmark for multi-view stereo reconstruction, we will use the Middlebury data sets, among others, for the evaluation of the reconstruction technique described in Chapter 4.



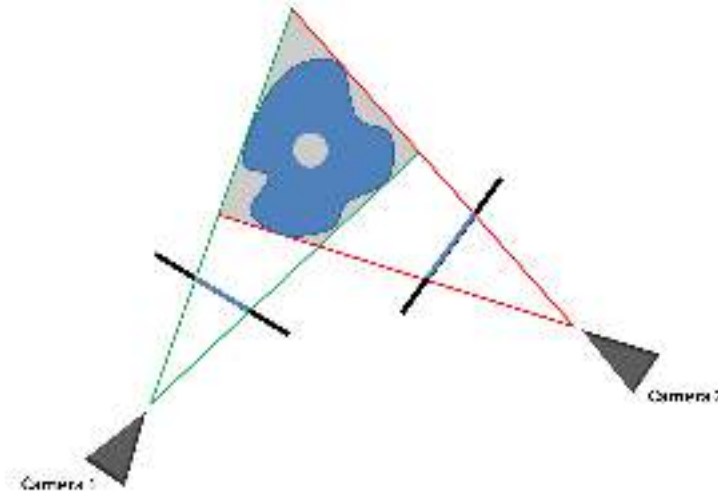
**Figure 1.4:** The Middlebury data sets. (a) Some images of the temple object (size:  $10\text{cm} \times 16\text{cm} \times 8\text{cm}$ ) and the dino object (size:  $7\text{cm} \times 9\text{cm} \times 7\text{cm}$ ). (b) Corresponding ground truth models captured by a laser stripe scanner. The images are obtained from the Middlebury webpage [46].

### 1.3.2 Reconstruction Approaches

The variety of multi-view stereo algorithms is vast. In what follows, we will distinguish certain types of reconstruction methods, highlight their advantages and disadvantages and present some outstanding techniques.

Many reconstruction approaches assume Lambertian surfaces, which means that the light is scattered in a way that the brightness of the surface is the same from every point of view. Hence, specular reflections are excluded. Consequently, if a 3D point lies on the object surface, its projection should have similar color values in each image in which it is visible. This constraint is known as *photo-consistency*. Therefore, 3D multi-view reconstruction techniques usually incorporate some kind of visibility information and an appropriate photo-consistency measure. Furthermore, some methods require a bounding volume that contains the object to be reconstructed. Assume that the object can be segmented as foreground regions (so-called *silhouettes*) in the input images. Then the silhouettes can be back-projected from the corresponding camera positions resulting in a set of cones. The intersection of all silhouette cones is called *visual hull* and contains the object, see [35]. It can, however, differ substantially from the actual shape since deep concavities and holes are not visible in the silhouettes. This behavior is indicated in Fig. 1.5, where the visual hull received from two images is sketched. Moreover, the segmentation of an object in several images might be time consuming or even impossible, e.g., for crowded scenes. Therefore, some reconstruction approaches avoid computing the visual hull by using alternative initial values while others do not need a bounding volume at all.

There are many different ways of representing 3D surfaces, yielding different conceptual types of 3D reconstruction methods. Similar to [48], we will distinguish four classes of techniques: feature based methods, methods based on discrete surface representations, methods based on continuous surface representations, and depth-map-based methods. In the following, we will describe the main concepts of each group. Subsequently, in Fig. 1.7, we will present examples of



**Figure 1.5:** The visual hull (gray) as the intersection of the silhouette cones (green and red) obtained from two images.

reconstruction results from the Middlebury evaluation [46]. Moreover, the runtimes as well as the accuracy and completeness values of the leading methods are presented in diagrams in Figs. 1.8 and 1.9 to quantitatively compare the different approaches. Since almost all authors of the state-of-the-art methods submitted their reconstructions for the “TempleRing” (47 views) and the “DinoRing” (48 views) data sets we will use the corresponding results for comparison. For the evaluation of some solutions for the smaller data sets “TempleSparse” (16 views) and “DinoSparse” (16 views) and the bigger ones “TempleFull” (312 views) and “DinoFull” (363 views) we refer to the Middlebury webpage [46].

**Feature-based methods:** Reconstruction algorithms based on features (e.g., [22], [25], [56]) usually represent the object surface by appropriate surface elements like planar disks [25] or rectangular patches [22]. They first define seed points by feature extraction and matching and then expand the surface. Using the visibility information gained from calibrated cameras, the resulting 3D shape can be refined in a third step. Contrary to other approaches, feature-based methods do not need any kind of prior information about the shape of the object or scene and therefore have few restrictions. On the Middlebury data-sets, the reconstructions of Furukawa et al. [22] are, in fact, among the best performers in terms of accuracy and completeness; see Figs. 1.8 and 1.9. Using an iterative deformation process, highly detailed surfaces are obtained, as can be seen in Fig. 1.7.

The good quality of feature-based reconstructions, however, also entails high computation times that can reach several hours. Therefore, these methods are not suitable for time-critical applications.

**Methods based on discrete surface representations:** This class of reconstruction techniques (e.g., [15], [29], [31], [33], [49], [50], [57]) uses a discretized 3D volume that contains the object to be reconstructed. The first approaches like voxel-coloring [49] and space carving [33] computed a photo-consistency measure on a voxel volume and then successively removed voxels with values below a certain threshold. These methods were purely local and no topological constraints or smoothness conditions could be incorporated. Consequently, they resulted in noisy

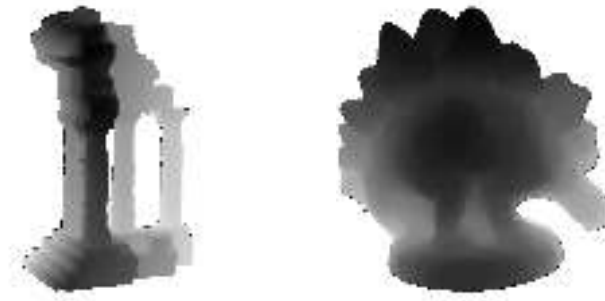


reconstructions and, therefore, more complex algorithms have been developed. In 2005 Vogiatzis et al. [57] introduced a reconstruction approach using volumetric graph cuts. Here a discrete cost functional  $E$  measuring photo-consistency is defined. The optimal surface minimizing  $E$  can then be found by discrete optimization. The voxels are embedded in a graph structure where every node represents a voxel, while the edges are weighted according to the cost functional. The optimal surface can then be found as minimal cut of the graph separating the interior and the exterior of the object. The algorithm of Vogiatzis et al. [57] and other graph-cut-based multi-view stereo algorithms [15], [29], [31], [50] yield accurate and complete results, see Figs. 1.8 and 1.9. Most of these methods, however, first compute a bounding volume containing the object surface. Often, the visual hull is used, which cannot always be determined or does not represent the object surface in a correct way, as mentioned above. Moreover, the resolution of the discretization influences accuracy, computation time, and memory. The bigger an object is, the more voxels are needed for a detailed representation, and the more memory and time is needed. Additionally, most of the classical graph cut algorithms are difficult to parallelize. Recently, there have been, however, promising approaches that overcome some of these problems. Hiep et al. [29] succeeded in reconstructing large scale objects under uncontrolled conditions. Parts of the algorithm are implemented on a GPU, which yields reasonable runtimes. In addition, at the moment their result for the Middlebury “TempeRing” data set is leading in accuracy and completeness; see Fig. 1.8. The reconstructions are shown in Fig. 1.7. In 2011, Chang et al. [15] introduced a method running entirely on a GPU. They combined the representation of surfaces by surface elements with a graph structure and optimization. The quality and efficiency of the corresponding results (see Figs. 1.7, 1.8, and 1.9) show that graph-cut-based methods are able to produce high-quality reconstructions in runtimes of only a few minutes.

**Methods based on continuous surface representations:** Similar to graph-cut-based methods, continuous reconstruction algorithms (e.g., [2], [17], [21], [23], [28], [32], [44], [51], [62]) define a (continuous) cost functional that must be minimized. In a variational approach, the optimal surface is found by solving a minimization problem. Methods based on polygon meshes define an initial surface represented by a mesh that is then iteratively deformed to obtain an optimal surface. This can be done, for example, by snake evolution [28], [51]. Hernández et al. [28] use texture and silhouette information that are incorporated into a snake framework. Corresponding forces deform an initial surface represented by a triangle mesh until all mesh points have converged. The solution of methods that use snake evolution is generally limited by the topology of an initial mesh. Zaharescu et al. [62], however, introduced a mesh-based approach that is purely geometric and allows topological changes. Level-set approaches [21], [32], [44] model surfaces as the zero set of volumetric functions. An optimal solution minimizing a corresponding cost functional is derived by solving partial differential equations. Pons et al. [44] minimize the prediction error: an image taken by one camera is back-projected onto the current surface estimate and then projected to the other cameras, yielding predicted images. Then the similarity between the original image and the predicted images is measured. Using variational methods, this similarity is maximized for all input images and an optimal surface is obtained. This reconstruction method can also handle non-Lambertian scenes and produces high quality results on the Middlebury data sets; see Fig. 1.7. Kolev et al. [32] use photo-consistency to define a convex energy model. An optimal surface is derived by continuous global optimization. Most of the continuous methods need an initial surface as the starting value for the optimization process. If the initial guess (e.g., the visual hull) does not represent the object correctly, the reconstruction might fail. Moreover, a big discrepancy between starting value and actual object surface can result in slow convergence and often convergence to a globally optimal solution

cannot be guaranteed. Several approaches in this class of reconstruction methods produce high-quality results (see Figs. 1.8 and 1.9), but in general they cannot compete with the runtimes of graph-cut or depth-map-based algorithms.

**Depth-map-based methods:** This last group of methods uses *depth maps* for 3D surface modeling. Depth maps, also called depth images, range images, or 2.5D images, contain 3D information of a scene measured from a known point of view. For every pixel, the gray value represents the distance of the corresponding 3D surface point to a reference point, see Fig. 1.6. Depth maps can be computed by active methods like laser scanners or by means of two images of the same scene taken from calibrated cameras. The latter is called *binocular stereo* and many different algorithms have been developed to solve this stereo matching problem in order to obtain high-quality depth maps. We refer to the work of Scharstein et al. [47] for an overview and evaluation of two-view stereo algorithms.



**Figure 1.6:** Examples of depth maps for two objects. The gray value represents the distance to a given reference point.

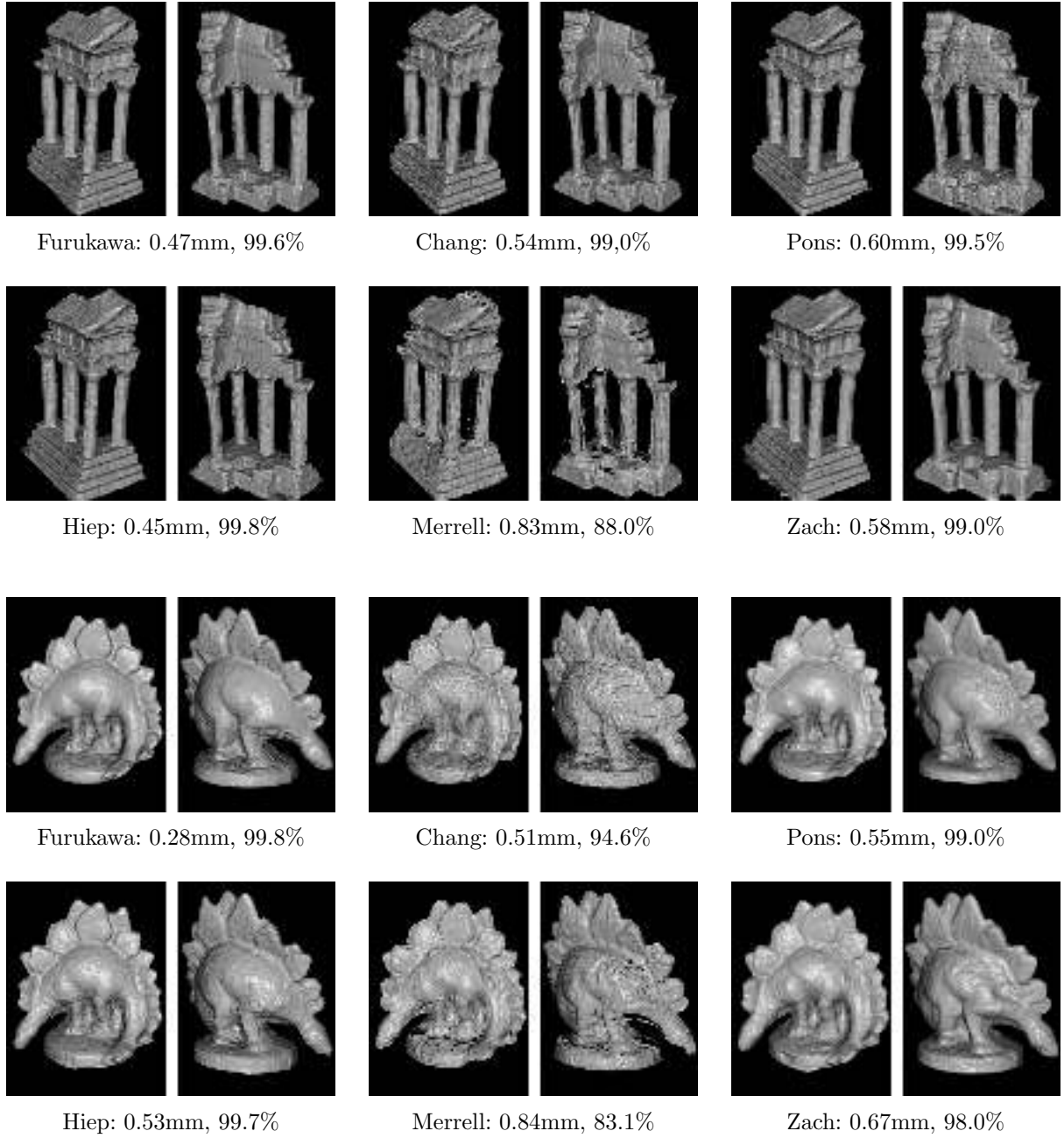
Multi-view stereo methods aim to reconstruct the whole surface of an object, not only the parts seen from certain positions. To achieve this, multiple depth maps created from different points of view are used. The reconstruction process can be described as follows: First, depth maps are computed from calibrated images. In a second step, these are then merged to obtain a 3D representation of the object.

Splitting the reconstruction into two parts implies that errors might propagate. At the same time, however, it provides a certain flexibility: The method used for constructing the depth maps can be chosen according to the object’s properties. While depth information for highly textured single objects can be obtained by simple and fast algorithms, treating objects with little texture or crowded scenes requires more complex techniques. By using an appropriate depth map estimation method, even the reconstruction of big objects and entire scenes becomes possible [40], [54]. Depth-map-based multi-view reconstruction algorithms can roughly be categorized into two groups: Approaches that concentrate on robust depth map estimation (e.g., [9], [39], [54]) and methods that focus on depth map fusion (e.g., [6], [18], [38], [40], [52], [59], [61]).

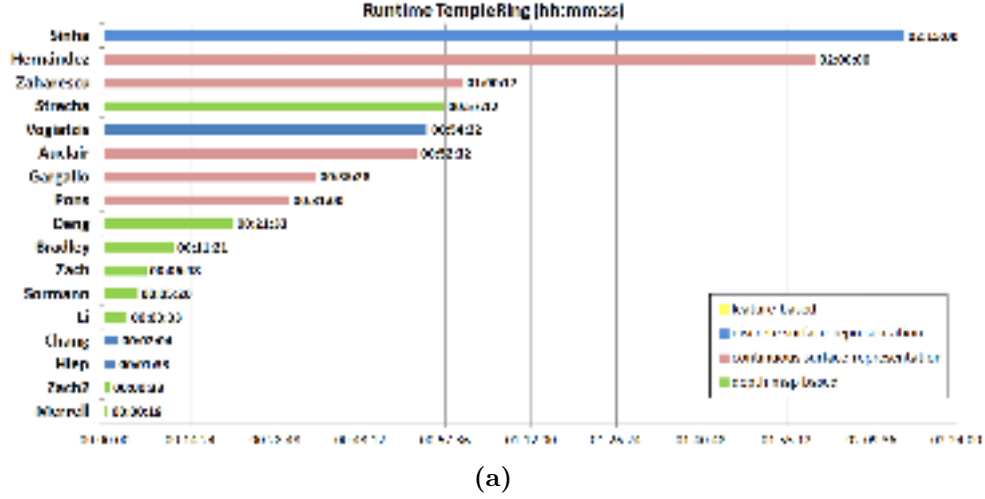
Algorithms in the first group aim to create high quality depth maps prior to the merging step. Campbell et al. [9] and Strecha et al. [54] use discrete Markov random field optimization to remove outliers from depth maps. Liu et al. [39] apply variational methods to create depth maps that are subsequently refined and eventually merged.

The second group of depth-map-based reconstruction methods assumes that depth maps have already been obtained by an appropriate stereo algorithm. These possibly noisy depth maps are then fused by robust techniques that remove outliers. There is a variety of concepts used for the merging stage: Li et al. [38] make use of a bundle optimization algorithm that yields accurate reconstructions and reasonable computation times. Sormann et al. [52] perform GPU-based depth estimation followed by volumetric fusion using graph cuts. One of the fastest reconstruction algorithms has been proposed by Merrell et al. [40]: A GPU-based implementation recursively merges depth maps from adjacent cameras. Reconstructions of big objects like entire buildings can be performed within seconds, but the corresponding 3D models lack accuracy, as can be seen in Fig. 1.7. Zach et al. ([59], [61]) introduced variational approaches that minimize convex energy functionals. The corresponding results are among the best performers on the Middlebury data sets, see Figs. 1.7, 1.8, and 1.9. In 2012, Deng et al. [18] suggested an algorithm that uses matrix completion, which until then was only known from signal processing.

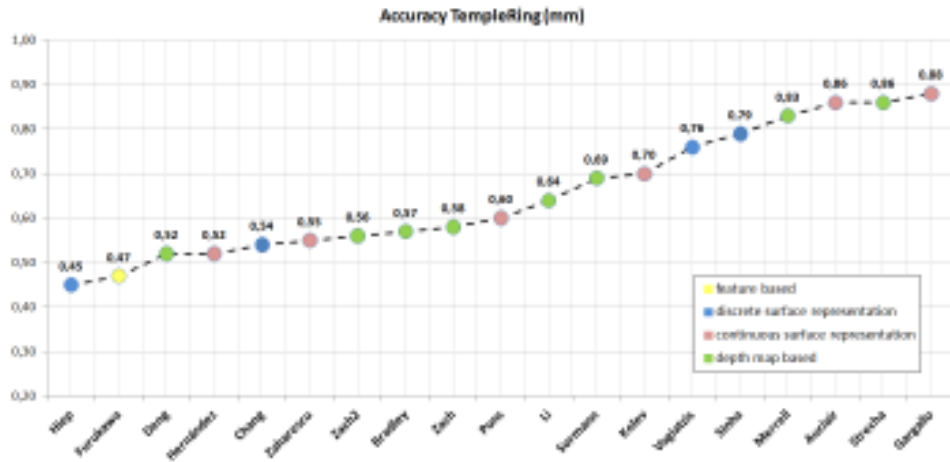
Many reconstruction methods based on depth maps can be at least partially parallelized and need low computation time, which makes them attractive for time-critical applications.



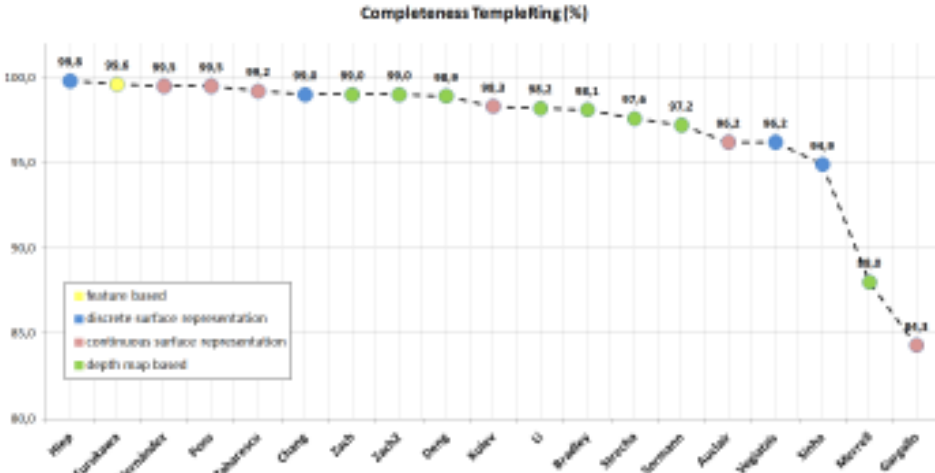
**Figure 1.7:** Surface reconstruction results for some of the state-of-the-art methods on the Middlebury data sets “TempleRing” (consisting of 47 images of size  $640 \times 480$ ) and “DinoRing” (consisting of 48 images of size  $640 \times 480$ ). For each reconstruction the accuracy and completeness values of the Middlebury evaluation are listed. Accuracy is specified as the distance  $d$  (in mm) such that 90% of the reconstruction lies within  $d$  of the ground truth model. Completeness is measured as the percent of points of the ground truth model that are within 1.25mm of the reconstruction. The images are obtained from the Middlebury webpage [46].



(a)

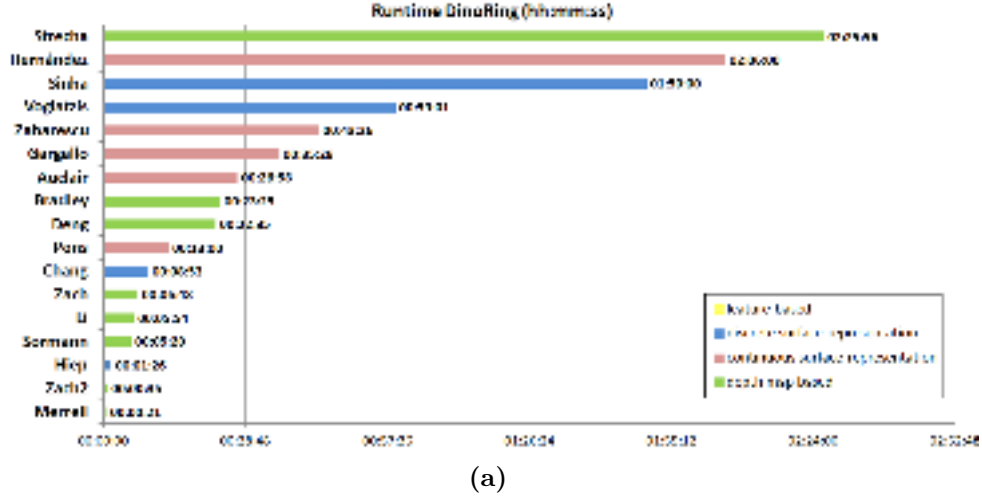


(b)

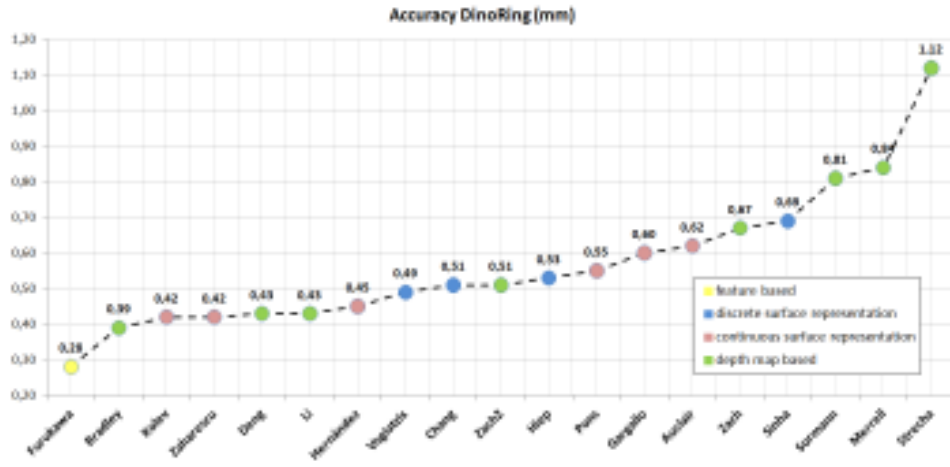


(c)

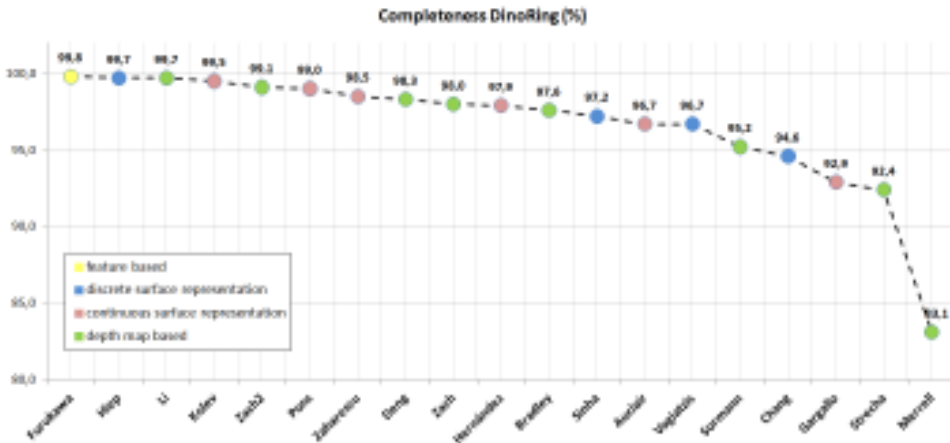
**Figure 1.8:** Quantitative comparison of several multi-view stereo reconstruction methods for the Middlebury “TempleRing” data set (47 views). (a) Runtimes (normalized to 3.0 Ghz). (b) Accuracy: Distance  $d$  (in mm) such that 90% of the reconstruction lie within  $d$  of the ground truth model. (c) Completeness: percent of points of the ground truth model that lie within 1.25mm of the reconstruction. The evaluation results are obtained from the Middlebury webpage [46].



(a)



(b)



(c)

**Figure 1.9:** Quantitative comparison of several multi-view stereo reconstruction methods for the Middlebury “DinoRing” data set (48 views). (a) Runtimes (normalized to 3.0 Ghz). (b) Accuracy: Distance  $d$  (in mm) such that 90% of the reconstruction lie within  $d$  of the ground truth model. (c) Completeness: percent of points of the ground truth model that lie within 1.25mm of the reconstruction. The evaluation results are obtained from the Middlebury webpage [46].

## 1.4 Selected Method

The choice of the right type of 3D reconstruction algorithm highly depends on the properties of the object or scene to be reconstructed and the subsequent applications. As demonstrated in Figs. 1.8a and 1.9a, currently only the methods based on graph cuts or depth maps are able to perform reconstructions in less than 30 minutes on the Middlebury data sets. Our aim was to implement a 3D reconstruction algorithm that can be used for different machine vision applications under different conditions. As mentioned above, depth map based approaches are very flexible due to the separation of depth map estimation and merging and are therefore applicable in various situations, which convinced us to choose one of these algorithms. Nevertheless, the graph-cut-based methods of Hiep et al. [29] and Chang et al. [15] should be mentioned since they produce promising results within reasonable runtimes. Considering the accuracy and completeness values of depth-map-based approaches in the Middlebury evaluation combined with the corresponding computation times and also the visual impression of the results, the method proposed by Zach et al. [61] and its accelerated version Zach2 [59] met our requirements best. In addition, the underlying mathematical theory is well established and can be used for similar computer vision problems.

Zach et al. developed a variational reconstruction method that concentrates on a robust merging of depth maps. A given set of depth maps  $r_i$  is first converted into three-dimensional distance fields  $f_i$  that contain the signed distances of 3D points to the object's surface. Then, an optimal distance field  $u$ , whose zero set represents the whole surface of the object or scene to be reconstructed, is obtained by solving the following variational problem

$$\min_{u \in X} J(u) + \lambda \int_{\Omega} \sum_{i \in I(x)} |u - f_i| dx. \quad (\text{Z})$$

Here,  $J$  denotes a special regularization term, the *total variation*, that ensures that the reconstructed surface is sufficiently smooth. The  $L^1$  data fidelity term  $\int_{\Omega} \sum_{i \in I(x)} |u - f_i| dx$ , however, yields a surface that is close to the surfaces implied by the input depth maps while removing outliers. To solve (Z), Zach et al. present a globally convergent optimization scheme.

In the following, we will treat this reconstruction algorithm in detail, analyze its properties, and show the convergence to an optimal surface. Moreover, implementation details will be provided and experimental results will be presented and discussed.

The outline is as follows: In Chapter 2, we will first present some mathematical background about convex functionals and optimization problems that will be used for the mathematical analysis of (Z). In Chapter 3, we will give a short introduction to variational problems in image analysis and define the *total variation* regularization. An appropriate discretization will be presented and some applications will be mentioned. Endowed with the mathematical theory, we will then (Chapter 4) discuss the structure and solvability of the reconstruction model (Z). To do so, we will first explain the construction of appropriate distance fields and then present the algorithm proposed by Zach et al. After showing some implementation details, we will demonstrate and discuss experimental results in Chapter 5. Finally (Chapter 6), possible improvements and conclusions will be addressed.





---

## Convex Functionals and Minimization Problems

---

As will be shown later, the energy functional in (Z) is convex, which yields some useful properties with respect to the optimization task. Each local minimum of a convex functional is also globally optimal, which facilitates the optimization process. Moreover, in convex optimization many algorithms make use of the duality concept: For a convex functional, a dual functional, the so-called *Fenchel conjugate*, is determined and a dual problem can be developed. From the dual solution, conclusions about the optimum of the original problem can be drawn.

In this chapter, we will recall some basic facts from convex analysis and optimization, which will be used for the mathematical analysis of problem (Z) later on. Starting with some general characteristics of *extended real-valued functionals*, that is functionals with values in  $\mathbb{R}_\infty = \mathbb{R} \cup \{\infty\}$ , we will introduce convex functionals and optimality conditions for convex optimization problems. Finally, the *Fenchel conjugate* will be defined and some of its most important properties will be presented.

Throughout this whole chapter, let  $X$  be a real Banach space and  $X^*$  the corresponding dual space.

### 2.1 Extended Real-Valued Functionals

In convex analysis, it is convenient to consider functionals  $F : X \rightarrow \mathbb{R}_\infty$  with

$$\mathbb{R}_\infty := \mathbb{R} \cup \{\infty\}$$

the set of extended real numbers. Hence, besides real values,  $F$  can also attain the value  $\infty$ . For a detailed description of  $\mathbb{R}_\infty$  including calculus rules, we refer to [30, Appendix 2]. We list some properties of extended real-valued functionals that are of particular importance when existence and uniqueness of a minimizer must be proved:

**Definition 2.1** (properties of functionals on  $\mathbb{R}_\infty$ )

An extended real-valued functional  $F : X \rightarrow \mathbb{R}_\infty$  is called

(i) **proper** if the **effective domain**  $\text{dom } F$ ,

$$\text{dom } F := \{u \in X : F(u) < \infty\}$$

is non-empty.

(ii) **lower semi-continuous**, if

$$F(u) \leq \liminf_{n \rightarrow \infty} F(u^n)$$

for any sequence  $(u^n) \subset X$  with  $\lim_{n \rightarrow \infty} u^n = u$ ,  $u \in X$ .

(iii) **coercive**, if

$$F(u^n) \rightarrow \infty \quad \text{as} \quad \|u^n\|_X \rightarrow \infty.$$

## 2.2 Convex Functionals

In this section, we present some important properties of convex functionals including the *sub-differential* and the *Fenchel conjugate*. For a detailed introduction to convex analysis, we refer to [30]. A summary of the most important characteristics of convex functionals related to variational problems in image processing can be found in [7].

**Definition 2.2** (convex, strictly convex functional)

A functional  $F : X \rightarrow \mathbb{R}_\infty$  is called **convex** if, for all  $u, v \in X$  and  $t \in [0, 1]$

$$F(tu + (1 - t)v) \leq tF(u) + (1 - t)F(v).$$

It is called **strictly convex** if, for all  $u, v \in X$ ,  $u \neq v$  and  $t \in (0, 1)$

$$F(tu + (1 - t)v) < tF(u) + (1 - t)F(v).$$

Because of its structure, convexity is preserved for certain combinations of convex functionals, which can be shown by simple calculations (see for example [7, Lemma 6.21]):

**Theorem 2.3** (composition of convex functionals)

Let  $F, G : X \rightarrow \mathbb{R}_\infty$  be convex. Then

(i)  $\alpha F$  is convex for all  $\alpha > 0$ .

(ii)  $F + G$  is convex.

(iii)  $\sup_{i \in I} F_i$  is convex for any family of convex functionals  $F_i : X \rightarrow \mathbb{R}_\infty$ .

Now, consider the general minimization problem

$$\min_{u \in X} F(u) \tag{2.1}$$

with  $F : X \rightarrow \mathbb{R}_\infty$ . A solution of (2.1) is called *minimizer* of  $F$ .

**Theorem 2.4** (local and global minimizer of convex functionals)

If  $F : X \rightarrow \mathbb{R}_\infty$  is convex, every local minimizer  $\bar{u} \in X$  is also a global minimizer. If  $F$  is strictly convex  $\bar{u}$  is unique.

**Proof:** Let  $\bar{u}$  be a local minimizer. Assume there is a  $u \in X$  with  $F(u) < F(\bar{u})$ . Then by convexity of  $F$  for all  $t \in [0, 1]$ :

$$F(tu + (1-t)\bar{u}) \leq tF(u) + (1-t)F(\bar{u}) < tF(\bar{u}) + (1-t)F(\bar{u}) = F(\bar{u})$$

what contradicts the local optimality of  $\bar{u}$ . Hence  $\bar{u}$  is a global minimizer.

Now, let  $F$  be strictly convex. Assume the existence of two minimizers  $\bar{u}$  and  $\bar{v}$ ,  $\bar{u} \neq \bar{v}$ . Then as shown above  $F(\bar{u}) = F(\bar{v})$  and for all  $t \in (0, 1)$

$$F(t\bar{u} + (1-t)\bar{v}) < tF(\bar{u}) + (1-t)F(\bar{v}) = F(\bar{u})$$

This contradicts the optimality of  $\bar{u}$ . Consequently  $\bar{u}$  is unique.  $\square$

But when does a local (= global) minimizer exists at all? An optimality condition for convex minimization problems can be derived using the so-called *subdifferential*, as described next.

### 2.2.1 The Subdifferential

For optimization problems with differentiable functions there are several optimality conditions that make use of derivatives. For smooth functions  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , a necessary condition for an extremum  $\bar{x}$  is  $\nabla f(\bar{x}) = 0$ .

In the following chapters, however, we will treat convex functionals that are not necessarily differentiable. The *subgradient* is an appropriate generalization of the derivative and the set of all subgradients, the *subdifferential* can be used to derive optimality conditions for general convex functionals.

**Definition 2.5** (subgradient, subdifferential)

Let  $F : X \rightarrow \mathbb{R}_\infty$  be convex.  $w \in X^*$  is called **subgradient** of  $F$  in  $u$ , if

$$F(u) + \langle w, v - u \rangle_{X^* \times X} \leq F(v) \quad \text{for all } v \in X.$$

The set

$$\partial F(u) = \{w \in X^* : w \text{ is a subgradient of } F \text{ in } u\}$$

is called **subdifferential** of  $F$  in  $u$ .

For convenience, we will write  $\langle \cdot \rangle$  for  $\langle \cdot \rangle_{X^* \times X}$  throughout this whole chapter.

Similar to ordinary derivatives, there are certain calculus rules for subdifferentials, which can be found in [7, Theorem 6.51] and [30, Corollary 2.1.4]. Here, we only mention three of them:

**Theorem 2.6** (properties of the subdifferential)

Let  $F, G : X \rightarrow \mathbb{R}_\infty$  be proper and convex. Then

- (i)  $\partial(\alpha F) = \alpha \partial F$  for  $\alpha > 0$ .
- (ii)  $\partial(F + G) \supset \partial F + \partial G$  and  $\partial(F + G) = \partial F + \partial G$  if  $F$  is continuous at some  $u^0 \in \text{dom } F \cap \text{dom } G$ .
- (iii) If  $F$  is differentiable  $\partial F(u) = \nabla F(u)$ .

Because of its characteristic structure, the subdifferential yields the following important optimality condition (see [7, Satz 6.43]), which is a generalization of the condition  $\nabla F(\bar{x}) = 0$  in the differentiable case.

**Theorem 2.7** (optimality condition for convex minimization)

Let  $F : X \rightarrow \mathbb{R}_\infty$  be proper and convex. Then  $\bar{u} \in X$  is a minimizer of  $F$  if and only if  $0 \in \partial F(\bar{u})$ .

**Proof:**  $\bar{u} \in X$  is a minimizer of  $F$  if and only if

$$F(\bar{u}) \leq F(u) \quad \forall u \in X. \quad (2.2)$$

Since

$$F(\bar{u}) = F(\bar{u}) + \langle 0, u - \bar{u} \rangle \quad \forall u \in X$$

the inequality in (2.2) is equivalent to

$$F(\bar{u}) + \langle 0, u - \bar{u} \rangle \leq F(u) \quad \forall u \in X$$

which is, by definition of the subdifferential,

$$0 \in \partial F(\bar{u}).$$

□

### 2.2.2 The Fenchel Conjugate

When treating optimization problems, it is often useful to consider the so-called *dual* optimization problem. This can be derived from the original problem, the *primal* one. Under certain conditions, a solution of the primal problem can be obtained by using the solution of its dual, which is a great advantage especially if the latter is easier to solve. For convex optimization problems, this duality is closely related to the *dual functional* also called *Fenchel conjugate*, which is treated in this section. We name some of its most important properties that will be used in the following chapters.

**Definition 2.8** (Fenchel conjugate, biconjugate)

Let  $F : X \rightarrow \mathbb{R}_\infty$ . The **Fenchel conjugate** of  $F$  is the functional  $F^* : X^* \rightarrow \mathbb{R}_\infty$ ,

$$F^*(w) = \sup_{u \in X} (\langle w, u \rangle - F(u)).$$

Accordingly  $F^{**} : X \rightarrow \mathbb{R}_\infty$ ,

$$F^{**}(u) = \sup_{w \in X^*} (\langle w, u \rangle - F^*(w)),$$

is called **biconjugate** of  $F$ .

Under certain conditions, the biconjugate equals the original functional ( $F = F^{**}$ ). We refer to [5, Chapter 4.2] for a detailed proof of the following theorem and some more facts about biconjugate functions.

**Theorem 2.9** (biconjugation)

Let  $F : X \rightarrow \mathbb{R}_\infty$  be proper, convex, and lower semi-continuous. Then  $F = F^{**}$ .

Because of its structure, the Fenchel conjugate is closely related to the subdifferential and therefore used for deriving optimality conditions.

**Theorem 2.10** (Fenchel inequality and conclusions)

For  $F : X \rightarrow \mathbb{R}_\infty$  with Fenchel conjugate  $F^* : X^* \rightarrow \mathbb{R}_\infty$  and  $u \in X, w \in X^*$ , the following statements hold:

(i) The **Fenchel inequality**

$$\langle w, u \rangle \leq F(u) + F^*(w) \quad (2.3)$$

holds.

(ii)  $w \in \partial F(u)$  if and only if  $\langle w, u \rangle = F(u) + F^*(w)$ .

(iii) If  $F$  is convex and lower semi-continuous then

$$w \in \partial F(u) \quad \text{if and only if} \quad u \in \partial F^*(w).$$

**Proof:**

(i) By definition

$$F^*(w) = \sup_{u \in X} (\langle w, u \rangle - F(u)).$$

Thus

$$F^*(w) \geq \langle w, u \rangle - F(u) \quad \text{for all } u \in X$$

and we get

$$\langle w, u \rangle \leq F(u) + F^*(w) \quad \text{for all } u \in X.$$

(ii) Let  $w \in \partial F(u)$ . Then, by definition of the subgradient

$$F(u) + \langle w, v - u \rangle \leq F(v) \quad \text{for all } v \in X.$$

Hence,

$$\langle w, u \rangle - F(u) \geq \langle w, v \rangle - F(v) \quad \text{for all } v \in X,$$

which means

$$\langle w, u \rangle - F(u) \geq F^*(w)$$

and we obtain

$$\langle w, u \rangle \geq F(u) + F^*(w).$$

Applying the Fenchel inequality (2.3) yields

$$\langle w, u \rangle = F(u) + F^*(w).$$

Otherwise, if  $\langle w, u \rangle = F(u) + F^*(w)$ , by definition of the Fenchel conjugate  $F^*$

$$\langle w, u \rangle = F(u) + \sup_{v \in X} (\langle w, v \rangle - F(v)).$$

Hence,  $F(u) < \infty$  and

$$\langle w, u \rangle \geq F(u) + \langle w, v \rangle - F(v) \quad \text{for all } v \in X.$$

This is equivalent to

$$F(u) + \langle w, v - u \rangle \leq F(v) \quad \text{for all } v \in X,$$

which means  $w \in \partial F(u)$ .

(iii) Let  $u \in \partial F^*(w)$ . Applying (ii) yields

$$u \in \partial F^*(w) \quad \text{if and only if} \quad \langle w, u \rangle = F^*(w) + F^{**}(u).$$

Since  $F$  is convex and lower semi-continuous, according to Theorem 2.9  $F^{**} = F$ , and we obtain

$$u \in \partial F^*(w) \quad \text{if and only if} \quad \langle w, u \rangle = F(u) + F^*(w).$$

Using (ii) again, it follows

$$u \in \partial F^*(w) \quad \text{if and only if} \quad w \in \partial F(u).$$

□

To conclude this section, we consider a special class of functionals and their Fenchel conjugates (see [7], Aufgabe 6.21):

**Definition 2.11** (positive  $p$ -homogeneous functional)

A functional  $F : X \rightarrow \mathbb{R}_\infty$  is called **positive  $p$ -homogeneous** with  $p \in [1, \infty[$  if

$$F(\alpha u) = \alpha^p F(u) \quad \text{for all } u \in X \text{ and } \alpha > 0.$$

**Definition 2.12** (indicator functional)

For a set  $K \subset X$  the **indicator functional**  $I_K : X \rightarrow \{0, \infty\}$  is defined as

$$I_K(u) = \begin{cases} 0 & \text{if } u \in K \\ \infty & \text{else} \end{cases}.$$

**Theorem 2.13** (Fenchel Conjugate of a one-homogeneous functional)

Let  $F : X \rightarrow \mathbb{R}_\infty$  be proper and positive one-homogeneous. Then  $F^* = I_K$  for a closed convex set  $K \subset X^*$ .

**Proof:** Let  $F : X \rightarrow \mathbb{R}_\infty$  be one-homogeneous with Fenchel conjugate  $F^*$ . Then the Fenchel inequality (2.3)

$$F^*(w) \geq \langle w, u \rangle - F(u) \quad \forall u \in X$$

holds for  $w \in X^*$ . Let  $\alpha > 0$ , then  $\alpha u \in X \quad \forall u \in X$ , and it follows

$$F^*(w) \geq \langle w, \alpha u \rangle - F(\alpha u) \quad \forall \alpha > 0,$$

which is

$$F^*(w) \geq \alpha (\langle w, u \rangle - F(u)) \quad \forall \alpha > 0$$

since  $F$  is one-homogeneous. Thus either  $F^*(w) = \infty$  or  $\langle w, u \rangle - F(u) \leq 0 \quad \forall u \in X$  and  $F^*(w) = 0$ . This means

$$F^*(w) = \begin{cases} 0 & \text{for } w \in K := \{q \in X^* : \langle q, u \rangle \leq F(u) \quad \forall u \in X\} \\ \infty & \text{otherwise} \end{cases}.$$

The set  $K$  is closed in  $X^*$  and for  $t \in [0, 1]$ ,  $w, q \in K$  and all  $u \in X$

$$\begin{aligned} \langle tw + (1-t)q, u \rangle &= t \langle w, u \rangle + (1-t) \langle q, u \rangle \\ &\leq tF(u) + (1-t)F(u) \\ &= F(u). \end{aligned}$$

Hence,  $tw + (1-t)q \in K$ , which means  $K$  is convex. □

---

## Total Variation in Image Processing

---

As mentioned in Section 1.4, this work presents a *variational* approach for reconstructing 3D surfaces. In general, variational methods define an energy functional  $E$  for a set of (extended) real-valued functions and search for extrema of  $E$ . In 1992 Rudin, Osher, and Fatemi [45] presented a variational approach for solving the image denoising problem, which was pioneering in the field of variational image processing. They introduced the so-called *total variation* regularization term, which since then has been used for many different applications in image processing. As indicated Section 1.4, Zach et al. [61] incorporated the total variation into an energy functional used for the reconstruction of 3D surfaces. The structure of 3D volumes is similar to that of 2D images. In the discrete setting, instead of pixels we have voxels, instead of image matrices we have volume tensors. Thus, 3D volumes can be seen as three-dimensional images. Even in the continuous setting, when treating images as functions in appropriate spaces, the extension to higher dimensions is straightforward. Because of this, the ideas of several image processing methods and the image denoising problem in particular can also be used for three-dimensional or, more generally,  $n$ -dimensional problems.

In this chapter, we will treat  $n$ -dimensional images and explain the idea of variational approaches to solve image processing problems. Furthermore, an appropriate function space for images and the total variation functional will be introduced. The variational denoising problem proposed by Rudin, Osher, and Fatemi will be discussed and an algorithm that returns a global solution will be presented. Finally, some other applications of total variation in image processing will be mentioned.

### 3.1 Variational Methods in Image Processing

There are several ways of representing images in a mathematical framework. Chan et. al [14, Chapter 3] describe some of them, like functions, wavelets, Markov random fields and level-sets. Which representation to use depends on the actual problem to solve. Many variational image processing methods define images as functions  $u : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ . For some applications, it is appropriate to only admit images with certain properties. Consider, for example, the problem of removing noise. Here an optimal solution should be an image that is preferably smooth while still

preserving sharp edges that describe object contours. Using a variational approach, conditions like these can be described by functionals that attain low values for images that fit into the desired model. Finding the optimal image is then equivalent to solving a minimization problem.

### 3.1.1 A Continuous Image Model

We will treat images as  $L^p$ -functions on a Lipschitz domain  $\Omega \subset \mathbb{R}^n$ ,  $n > 0$ . Recall for  $p \in [1, \infty)$  and  $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$  the Lebesgue space

$$L^p(\Omega) = \left\{ u : \Omega \rightarrow \overline{\mathbb{R}} : \int_{\Omega} |u(x)|^p dx < \infty \right\}$$

with norm

$$\|u\|_{L^p(\Omega)} = \left( \int_{\Omega} |u(x)|^p dx \right)^{\frac{1}{p}}.$$

Hence, an image can be described as a function  $u \in L^p(\Omega)$ . Usually, two-dimensional gray-valued images are defined on  $\Omega \subset \mathbb{R}^2$  with values in  $[0, 1]$ .

**Remark:** If not noted otherwise, in the following the word *image* describes a general  $n$ -dimensional image. Thus, we do not restrict ourselves to the two-dimensional case.

### 3.1.2 Variational Problems

Variational approaches for image processing problems search among all possible images the one that minimizes a certain energy functional. In mathematical terms, one wants to solve the optimization problem

$$\min_{u \in X} E(u) = \min_{u \in X} \Psi(u) + \lambda \Phi(u, v) \quad (3.1)$$

with  $\lambda > 0$ .  $X$  denotes the function space containing all admitted images  $u$  and  $v \in Y$  is the given data.  $\Psi$  is called the *regularization* term and ensures that an optimal solution  $\bar{u} \in X$  (if it exists) satisfies certain properties.  $\Phi$  is called the *data fidelity* term: a possible solution should match the given data  $v$ , yielding low values for  $\Phi(u, v)$ .

For any such problem, one must consider the following questions:

- What kind of data is given? How to choose the data fidelity term?
- What (spatial) properties should a possible solution have? How to choose the regularization term?
- When does a solution exist?
- When is this solution unique?

### 3.1.3 Variational Denoising

The standard example for these kinds of problems is variational image denoising; see Fig. 3.1: Given a noisy image  $v$ , one wants to recover a denoised version that fulfills certain regularization properties. The data fidelity term  $\Phi(u) = \int_{\Omega} (u(x) - v(x))^2 dx$  makes sure that an optimal solution is similar to the given data. Another choice for  $\Phi$  could be, for example, the functional  $\int_{\Omega} |u(x) - v(x)| dx$ , which is less sensitive to outliers.





**Figure 3.1:** Variational image denoising. (a) Noisy input image. (b) Denoised image.

But how to choose  $\Psi$ ? How should an optimal solution look like? We want to get a *smoothed* version  $u$  of the given image  $v$ . Hence, the regularization term should perform a spatial regularization and penalize the image parts that are not smooth. One way to measure smoothness is the image gradient  $\nabla u$ . It attains high values at discontinuities while being low for homogeneous regions. The major problem, however, is that not only noisy parts but also edges present discontinuities and yield high gradient values. Thus, it is difficult to preserve sharp features while smoothing the image. Therefore, as shown in [11, Section 1.1.2], the possible choice  $\Psi(u) = \int_{\Omega} |\nabla u(x)|^2 dx$  is not desirable since edges cannot be retained because of too much regularization. To solve this problem, consider the functional  $\Psi(u) = \int_{\Omega} |\nabla u(x)| dx$ , which penalizes edges less. Together with the quadratic data fidelity term described above, we then obtain the optimization problem

$$\min_{u \in X} \int_{\Omega} |\nabla u(x)| dx + \lambda \int_{\Omega} (u(x) - v(x))^2 dx \quad (3.2)$$

The energy functional in (3.2) is well defined for functions in  $X \subset W^{1,1}(\Omega)$ , i.e., functions that are weakly differentiable and whose derivative is one-integrable. For a detailed introduction to Sobolev spaces, we refer to [20, Chapter 5]. It can be shown (see, for example, [11, Section 1.1.3]), however, that a function in  $W^{1,1}(\Omega)$  cannot have discontinuities across hypersurfaces and that in general there is no solution for (3.2) in  $W^{1,1}(\Omega)$ . This is why in 1992, Rudin, Osher, and Fatemi [45] suggested to use an extended version of the regularization term  $\int_{\Omega} |\nabla u(x)| dx$  for the image denoising problem. It admits a more general class of functions (images) as described next.

## 3.2 Total Variation in the Continuous Setting

Rudin, Osher, and Fatemi proposed a variational denoising model, which is known as “ROF model” and can be written as (see [12])

$$\min_{u \in L^2(\Omega)} J(u) + \frac{1}{2\theta} \int_{\Omega} (u(x) - v(x))^2 dx. \quad (\text{ROF})$$

with  $v \in L^2(\Omega)$ ,  $\theta > 0$  and the *total variation*  $J$  defined as follows:

**Definition 3.1** (total variation)

Let  $\Omega$  be an open subset of  $\mathbb{R}^n$ ,  $u \in L^1(\Omega)$ . Then the **total variation** (TV)  $J$  is defined as

$$J(u) = \sup \left\{ \int_{\Omega} u(x) \operatorname{div} \varphi(x) dx : \varphi \in C_c^\infty(\Omega, \mathbb{R}^n), |\varphi(x)| \leq 1 \quad \forall x \in \Omega \right\}, \quad (\text{TV})$$

where  $C_c^\infty(\Omega, \mathbb{R}^n)$  is the space of compactly supported functions  $\varphi : \Omega \rightarrow \mathbb{R}^n$  that are infinitely differentiable.

**Remark:** Functions  $u \in L^1(\Omega)$  with  $J(u) < \infty$  are called **functions with bounded variation** and together with the norm  $\|u\|_{BV(\Omega)} = \|u\|_{L^1(\Omega)} + J(u)$  they form the Banach space  $BV(\Omega)$ . We refer to [14] for an introduction to  $BV$ -functions and their application in image processing and analysis. A more detailed treatment of the space  $BV(\Omega)$  can be found in [1].

Consider the Sobolev space  $W^{1,1}(\Omega)$ . It can be shown (see, for example, [14, Chapter 2.2.1]) that  $W^{1,1}(\Omega) \subset BV(\Omega)$  on a Lipschitz domain  $\Omega \subset \mathbb{R}^n$  and

$$J(u) = \int_{\Omega} |\nabla u| dx \quad \text{for all } u \in W^{1,1}(\Omega).$$

Thus,  $J$  generalizes  $\int_{\Omega} |\nabla u| dx$ , as intended.

The total variation satisfies some important properties that enable conclusions about the solvability of minimization problems like (ROF):

**Theorem 3.2** (properties of the total variation)

The total variation  $J$  defined in (TV) is

- (i) non-negative,
- (ii) proper,
- (iii) convex,
- (iv) lower semi-continuous, and
- (v) one-homogeneous.

**Proof:** Similar to [11, Chapter 1.2.3], we define  $J_\varphi : L^p(\Omega) \rightarrow \mathbb{R}$ ,  $p \in [1, \infty[$  as

$$J_\varphi(u) = \int_{\Omega} u(x) \operatorname{div} \varphi(x) dx.$$

Then the total variation can be written as

$$J(u) = \sup_{\varphi \in T} J_\varphi(u)$$

with

$$T = \{ \varphi \in C_c^\infty(\Omega, \mathbb{R}^N), |\varphi(x)| \leq 1 \quad \forall x \in \Omega \}.$$

$J_\varphi$  is a linear operator and by Hölder's inequality

$$|J_\varphi(u)| = \left| \int_{\Omega} u(x) \operatorname{div} \varphi(x) dx \right| \leq \int_{\Omega} |u(x) \operatorname{div} \varphi(x)| dx \leq \|u\|_{L^p(\Omega)} \|\operatorname{div} \varphi\|_{L^q(\Omega)}$$

for  $q \in [1, \infty]$  such that  $\frac{1}{p} + \frac{1}{q} = 1$ . Hence,  $J_\varphi$  is bounded.

- (i) Since  $0 \in C_c^\infty(\Omega, \mathbb{R}^n)$ , by definition  $J(u) \geq J(0) = 0$ .
- (ii) (i) implies  $\text{dom } J \neq \emptyset$ , which means  $J$  is proper.
- (iii) Let  $u, v \in \Omega$ ,  $t \in [0, 1]$ . By linearity of  $J_\varphi$  it follows

$$\begin{aligned}
 J(tu + (1-t)v) &= \sup_{\varphi \in T} J_\varphi(tu + (1-t)v) \\
 &= \sup_{\varphi \in T} [tJ_\varphi(u) + (1-t)J_\varphi(v)] \\
 &\leq t \sup_{\varphi \in T} J_\varphi(u) + (1-t) \sup_{\varphi \in T} J_\varphi(v) \\
 &= tJ(u) + (1-t)J(v),
 \end{aligned}$$

that is,  $J$  is convex.

- (iv) Let  $(u^n) \subset L^p(\Omega)$ ,  $p \in [1, \infty[$  with  $u^n \rightharpoonup u$  in  $L^p(\Omega)$ . Then  $J_\varphi(u^n) \rightarrow J_\varphi(u)$  as  $n \rightarrow \infty$  since  $J_\varphi$  is linear and bounded. Hence,

$$\begin{aligned}
 J_\varphi(u) &= \lim_{n \rightarrow \infty} J_\varphi(u^n) \\
 &= \liminf_{n \rightarrow \infty} J_\varphi(u^n) \\
 &\leq \liminf_{n \rightarrow \infty} J(u^n)
 \end{aligned}$$

Taking the supremum over all  $\varphi \in T$  we obtain

$$J(u) \leq \liminf_{n \rightarrow \infty} J(u^n),$$

which means  $J$  is lower semi-continuous.

- (v) Let  $\alpha > 0$ . Then

$$J(\alpha u) = \sup_{\varphi \in T} J_\varphi(\alpha u) = \alpha \sup_{\varphi \in T} J_\varphi(u) = \alpha J(u).$$

Thus,  $J$  is one-homogeneous.

□

**Theorem 3.3** (unique solution of the image denoising problem (ROF))  
*The optimization problem (ROF) has a unique solution in  $BV(\Omega)$ .*

**Proof:** Similar to [11, Section 1.2.3], we define

$$E_{ROF}(u) = J(u) + \frac{1}{2\theta} \int_{\Omega} (u(x) - v(x))^2 dx. \quad (3.3)$$

**Existence:** According to Theorem 3.2 (i),  $J(u) \geq 0$ , and it follows that  $E_{ROF}(u) \geq 0$ , which means  $E_{ROF}$  is bounded. Hence, there exists a minimizing sequence  $(u^n) \subset L^2(\Omega)$  with

$$\lim_{n \rightarrow \infty} E_{ROF}(u^n) = \inf_{u \in L^2(\Omega)} E_{ROF}(u).$$

Furthermore, for  $n$  large enough  $E_{ROF}(u^n) \leq E_{ROF}(0) < \infty$ , which means  $(u^n)$  is bounded. It follows, that there exists a subsequence  $(u^{n_k})$  with  $u^{n_k} \rightharpoonup \bar{u}$  in  $L^2(\Omega)$ . Since  $J$  is lower semi-continuous, according to Theorem 3.2 (iii) and

$$\int_{\Omega} (\bar{u}(x) - v(x))^2 dx \leq \liminf_{k \rightarrow \infty} \int_{\Omega} (u^{n_k}(x) - v(x))^2 dx$$

we obtain

$$\inf_{u \in L^2(\Omega)} E_{ROF}(u) \leq E_{ROF}(\bar{u}) \leq \lim_{k \rightarrow \infty} \inf E_{ROF}(u^{n_k}) = \inf_{u \in L^2(\Omega)} E_{ROF}(u).$$

Thus,  $\bar{u}$  is a minimizer of  $E_{ROF}$ .

**Uniqueness:** Assume there exists another solution  $\hat{u} \in BV(\Omega)$ ,  $\bar{u} \neq \hat{u}$ . Then, since  $J$  is convex (see Theorem 3.2 (ii))

$$\begin{aligned} E_{ROF}\left(\frac{\bar{u} + \hat{u}}{2}\right) &= J\left(\frac{\bar{u} + \hat{u}}{2}\right) + \frac{1}{2\theta} \int_{\Omega} \left(\frac{\bar{u} + \hat{u}}{2} - v\right)^2 dx \\ &\leq \frac{1}{2}J(\bar{u}) + \frac{1}{2}J(\hat{u}) + \frac{1}{2\theta} \int_{\Omega} \left(\frac{\bar{u} + \hat{u}}{2} - v\right)^2 dx \\ &= \frac{1}{2}J(\bar{u}) + \frac{1}{2}J(\hat{u}) + \frac{1}{4\theta} \int_{\Omega} (\bar{u} - v)^2 dx + \frac{1}{4\theta} \int_{\Omega} (\hat{u} - v)^2 dx - \frac{1}{8\theta} \int_{\Omega} (\bar{u} - \hat{u})^2 dx \\ &= \frac{1}{2}E_{ROF}(\bar{u}) + \frac{1}{2}E_{ROF}(\hat{u}) - \frac{1}{8\theta} \int_{\Omega} (\bar{u} - \hat{u})^2 dx, \end{aligned}$$

which would be strictly less than  $\inf_{u \in X} E_{ROF}(u)$  if  $\bar{u} \neq \hat{u}$ . Thus,  $\bar{u} = \hat{u}$  and the solution is unique.  $\square$

Since Rudin, Osher, and Fatemi introduced the model (ROF) many different methods to solve the denoising problem have been elaborated. Chambolle [10] developed an efficient algorithm that is based on a dual formulation of (ROF). Since it is formulated in the discrete setting, in the following, we will present a discretization of the total variation. In Section 3.4, we will then derive the dual of (ROF) and present the algorithm suggested by Chambolle.

### 3.3 Total Variation in the Discrete Setting

Since the input of image processing applications are discrete images, methods developed in the continuous setting must be discretized. We will now introduce a discretization of the total variation. Therefore, we first consider a discrete image model.

#### 3.3.1 Discrete Image Model

For the discrete case, we only consider two-dimensional and three-dimensional images. The extension to higher dimensions, however, is straightforward.

- **2D images:** We define discrete 2D image as signals  $u : \{1, \dots, N\} \times \{1, \dots, M\} \rightarrow \mathbb{R}$  where  $N > 0$  and  $M > 0$  denote the image height and width, respectively. They can be represented as  $N \times M$  matrices  $(u_{i,j})_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M}} \in \mathbb{R}^{N \times M}$ .
- **3D image:** Analogously to the two dimensional case, 3D images (volumes) are defined as signals  $u : \{1, \dots, N\} \times \{1, \dots, M\} \times \{1, \dots, L\} \rightarrow \mathbb{R}$  where  $N > 0$ ,  $M > 0$  and  $L > 0$  denote the volume height, width and depth, respectively. They can be represented as  $N \times M \times L$  tensors  $(u_{i,j,k})_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M \\ 1 \leq k \leq L}} \in \mathbb{R}^{N \times M \times L}$ .

In [10], Chambolle presents a discretization of the total variation in the two-dimensional case. The multi-view 3D reconstruction algorithm described in the next chapter, however, works with 3D volumes. Therefore, we will introduce a discretization of the three-dimensional total variation in direct analogy to [10].

### 3.3.2 Discretization of the Total Variation

Let  $X$  be the Euclidean space  $\mathbb{R}^{N \times M \times L}$  with the inner product

$$(u, v)_X = \sum_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M \\ 1 \leq k \leq L}} u_{i,j,k} \cdot v_{i,j,k}$$

and norm

$$\|u\|_X = \sqrt{(u, u)_X}.$$

For a volume  $u \in X$ , we define the discrete total variation as

$$J(u) = \sum_{i,j,k} |(\nabla u)_{i,j,k}|$$

with the discrete gradient  $\nabla u \in Z := X \times X \times X$  given by

$$\begin{aligned} (\nabla u)_{i,j,k} &= ((\nabla u)_{i,j,k}^1, (\nabla u)_{i,j,k}^2, (\nabla u)_{i,j,k}^3), \\ (\nabla u)_{i,j,k}^1 &= \begin{cases} u_{i+1,j,k} - u_{i,j,k} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \\ (\nabla u)_{i,j,k}^2 &= \begin{cases} u_{i,j+1,k} - u_{i,j,k} & \text{if } j < M \\ 0 & \text{if } j = M \end{cases} \\ (\nabla u)_{i,j,k}^3 &= \begin{cases} u_{i,j,k+1} - u_{i,j,k} & \text{if } k < L \\ 0 & \text{if } k = L \end{cases} \end{aligned}$$

and the Euclidean norm  $|y| = \sqrt{y_1^2 + y_2^2 + y_3^2}$  for  $y = (y_1, y_2, y_3) \in \mathbb{R}^3$ . For a detailed treatment of discretizations of the total variation as approximations of the continuous one introduced in (TV), we refer to [34] and [58].

The discretization of (ROF) then reads

$$\min_{u \in X} J(u) + \frac{1}{2\theta} \|u - v\|_X^2. \quad (\text{ROF}_d)$$

For later use, we also consider the adjoint  $\nabla^*$  of the discrete gradient  $\nabla$ , which is defined by

$$\langle \nabla u, p \rangle_Z = \langle u, \nabla^* p \rangle_X$$

for all  $u \in X$ ,  $p = (p^1, p^2, p^3) \in Z$  with the inner product

$$(p, q)_Z = \sum_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M \\ 1 \leq k \leq L}} p_{i,j,k}^1 q_{i,j,k}^1 + p_{i,j,k}^2 q_{i,j,k}^2 + p_{i,j,k}^3 q_{i,j,k}^3. \quad (3.4)$$

Analogously to the continuous case (see [7, Satz 6.88]), let  $\operatorname{div} = -\nabla^*$ . Then

$$\langle \nabla u, p \rangle_Z = -\langle u, \operatorname{div} p \rangle_X$$

and it can be shown that

$$\begin{aligned} (\operatorname{div} p)_{i,j,k} = & \begin{cases} p_{i,j,k}^1 - p_{i-1,j,k}^1 & \text{if } 1 < i < N \\ p_{i,j,k}^1 & \text{if } i = 1 \\ -p_{i-1,j,k}^1 & \text{if } i = N \end{cases} \\ & + \begin{cases} p_{i,j,k}^2 - p_{i,j-1,k}^2 & \text{if } 1 < j < M \\ p_{i,j,k}^2 & \text{if } j = 1 \\ -p_{i,j-1,k}^2 & \text{if } j = M \end{cases} \\ & + \begin{cases} p_{i,j,k}^3 - p_{i,j,k-1}^3 & \text{if } 1 < k < L \\ p_{i,j,k}^3 & \text{if } k = 1 \\ -p_{i,j,k-1}^3 & \text{if } k = L \end{cases}. \end{aligned}$$

### 3.4 An Algorithm to Solve the Variational Denoising Problem

In 2004, Chambolle [10] used a dual approach and derived a globally convergent algorithm to solve the image denoising problem in the two-dimensional case. Because we deal with 3D volumes, we extended the method to a higher dimension. The resulting three-dimensional equivalent to [10] is described next.

#### 3.4.1 Derivation of the Dual Formulation

Consider the minimization problem

$$\min_{u \in L^2(\Omega)} J(u) + \int_{\Omega} \frac{1}{2\theta} (u(x) - v(x))^2 dx \quad (\text{ROF})$$

in the continuous setting, where  $\Omega$  is an open subset of  $\mathbb{R}^3$ ,  $v \in L^2(\Omega)$  and  $\theta > 0$ . As above,

$$E_{\text{ROF}}(u) := J(u) + \int_{\Omega} \frac{1}{2\theta} (u(x) - v(x))^2 dx.$$

Then

$$E_{\text{ROF}}(u) = J(u) + \frac{1}{2\theta} \|u - v\|_{L^2}^2$$

and because of the convexity of the norm and the total variation (see Theorem 3.2 (iii))  $E_{\text{ROF}}$  is convex. Moreover,  $E_{\text{ROF}}(0) = \|v\|_{L^2(\Omega)}^2 / 2\theta < \infty$ . Thus,  $E_{\text{ROF}}$  is proper. Then, according to Theorem 2.7, a  $\bar{u} \in L^2(\Omega)$  minimizes  $E_{\text{ROF}}$  if and only if  $0 \in \partial E_{\text{ROF}}(\bar{u})$ . Using Theorem 2.6, we obtain

$$0 \in \partial E_{\text{ROF}}(\bar{u}) \quad \Leftrightarrow \quad 0 \in \partial J(\bar{u}) + \partial \left( \frac{1}{2\theta} \|u - v\|_{L^2}^2 \right)(\bar{u})$$

and further

$$0 \in \partial E_{\text{ROF}}(\bar{u}) \quad \Leftrightarrow \quad 0 \in \partial J(\bar{u}) + \frac{\bar{u} - v}{\theta} \quad \Leftrightarrow \quad \frac{v - \bar{u}}{\theta} \in \partial J(\bar{u}).$$

According to Theorem 3.2,  $J$  is convex and lower semi-continuous and with Theorem 2.10(iii) it follows that

$$\frac{v - \bar{u}}{\theta} \in \partial J(\bar{u}) \quad \Leftrightarrow \quad \bar{u} \in \partial J^* \left( \frac{v - \bar{u}}{\theta} \right).$$

Furthermore,

$$\bar{u} \in \partial J^* \left( \frac{v - \bar{u}}{\theta} \right) \quad \Leftrightarrow \quad 0 \in \frac{v - \bar{u}}{\theta} - \frac{v}{\theta} + \frac{1}{\theta} \partial J^* \left( \frac{v - \bar{u}}{\theta} \right).$$

Substituting  $\bar{w} := (v - \bar{u})/\theta$ , we obtain

$$0 \in \bar{w} - \frac{v}{\theta} + \frac{1}{\theta} \partial J^*(\bar{w}).$$

Using Theorem 2.6 and Theorem 2.7 again,  $\bar{u}$  solves (ROF) if and only if  $\bar{w}$  is a solution of

$$\min_w \frac{\|w - v/\theta\|^2}{2} + \frac{1}{\theta} J^*(w). \quad (3.5)$$

This is the dual formulation of (ROF). Given the solution  $\bar{w}$  of (3.5), the solution  $\bar{u}$  of the primal problem (ROF) can be obtained by

$$\bar{u} = v - \theta \bar{w}. \quad (3.6)$$

### 3.4.2 The Dual Problem

To derive an optimization algorithm for (3.5), first the Fenchel conjugate  $J^*(w)$  of the total variation must be determined: By definition,

$$J^*(w) = \sup_u (\langle u, w \rangle - J(u)),$$

with  $\langle \cdot, \cdot \rangle$  the inner product in the Hilbert space  $L^2(\Omega)$ . As shown in Theorem 3.2,  $J$  is convex and one-homogeneous. The Fenchel conjugate of an one-homogeneous convex functional is the characteristic function of a closed convex set  $K$  (see Theorem 2.13). Hence,

$$J^*(w) = \begin{cases} 0 & \text{if } w \in K \\ +\infty & \text{otherwise.} \end{cases} \quad (3.7)$$

To determine  $K$ , consider  $J^{**}$  defined by

$$J^{**}(u) = \sup_w (\langle w, u \rangle - J^*(w)).$$

Using (3.7), we obtain

$$J^{**}(u) = \sup_{w \in K} \langle w, u \rangle.$$

Further,  $J$  is convex, proper, and lower semi-continuous (see Theorem 3.2), which, according to Theorem 2.9, implies

$$J^{**} = J.$$

Since, by definition

$$J(u) = \sup \{ \langle u, \operatorname{div} \varphi \rangle : \varphi \in C_c^1(\Omega; \mathbb{R}^3), |\varphi(x)| \leq 1 \quad \forall x \in \Omega \}$$

$K$  is the closure of the set

$$\{\operatorname{div} \varphi : \varphi \in C_c^1(\Omega; \mathbb{R}^3), |\varphi(x)| \leq 1 \forall x \in \Omega\}.$$

Using (3.7), the dual optimization problem (3.5) reads

$$\min_{w \in K} \frac{\|w - v/\theta\|^2}{2}. \quad (3.8)$$

Thus, the solution  $\bar{w}$  of (3.8) is the unique projection of  $v/\theta$  on the closed convex set  $K$  denoted by  $P_K(v/\theta)$ , see [30, Chapter 3.1]. Consequently,  $\theta\bar{w} = P_{\theta K}(v)$  and using (3.6) we obtain

$$\bar{u} = v - P_{\theta K}(v). \quad (3.9)$$

Hence, solving (ROF) results in computing the projection  $P_{\theta K}(v)$ , that is, solving

$$\min_{w \in \theta K} \frac{\|w - v\|^2}{2}. \quad (3.10)$$

Because we work with discrete images, we will now formulate (3.10) in the discrete setting and present the algorithm proposed by Chambolle.

### 3.4.3 Discrete Formulation and Optimization

We consider images in the Euclidean space  $X = \mathbb{R}^{N \times M \times L}$ . Recall the discrete denoising problem

$$\min_{u \in X} J(u) + \frac{1}{2\theta} \|u - v\|_X^2. \quad (\text{ROF}_d)$$

To obtain a discrete formulation of the optimization problem (3.10) and thus the dual of  $(\text{ROF}_d)$ , we use the discretization of the volume gradient and divergence introduced in Chapter 3.3.2. Let  $Z = X \times X \times X$ . The discretization of the set  $K$  is given by

$$K = \{\operatorname{div} p : p \in Z, |p_{i,j,k}| \leq 1, \forall i = 1, \dots, N, j = 1, \dots, M, k = 1, \dots, L\}$$

and the discrete dual optimization problem can be written as

$$\min_{p \in Z} \|\theta \operatorname{div} p - v\|_X^2 \quad \text{such that} \quad |p_{i,j,k}|^2 - 1 \leq 0, \quad \forall i, j, k. \quad (\text{D})$$

Using the Karush-Kuhn-Tucker conditions for constrained convex minimization problems (see [30, Theorem 2.1.4]),  $p \in Z$  solves (D) if there exist Lagrangian multipliers  $\mu_{i,j,k} \geq 0$  with

$$-(\nabla(\theta \operatorname{div} p - v))_{i,j,k} + \mu_{i,j,k} \cdot p_{i,j,k} = 0 \quad \forall i, j, k, \quad (3.11)$$

where

$$\mu_{i,j,k} (|p_{i,j,k}|^2 - 1) = 0.$$

This means, either

$$\mu_{i,j,k} = 0 \quad \text{and} \quad |p_{i,j,k}|^2 \leq 1 \quad (3.12)$$

or

$$\mu_{i,j,k} \geq 0 \quad \text{and} \quad |p_{i,j,k}|^2 = 1 \quad (3.13)$$



Combining (3.12) with (3.11), we obtain

$$(\nabla(\theta \operatorname{div} p - v))_{i,j,k} = 0 = \mu_{i,j,k}$$

and combining (3.13) with (3.11) yields

$$\mu_{i,j,k} = |(\nabla(\theta \operatorname{div} p - v))_{i,j,k}|.$$

Thus, in both cases

$$\mu_{i,j,k} = |(\nabla(\theta \operatorname{div} p - v))_{i,j,k}| \quad (3.14)$$

and, using (3.14) in (3.11), it follows that

$$-(\nabla(\theta \operatorname{div} p - v))_{i,j,k} + |(\nabla(\theta \operatorname{div} p - v))_{i,j,k}| \cdot p_{i,j,k} = 0 \quad \forall i, j, k. \quad (3.15)$$

To solve (3.15), Chambolle [10] suggests the following fixed point scheme:

$$p_{i,j,k}^{n+1} = p_{i,j,k}^n + \tau \left( (\nabla(\theta \operatorname{div} p^n - v))_{i,j,k} - |(\nabla(\theta \operatorname{div} p^n - v))_{i,j,k}| \cdot p_{i,j,k}^{n+1} \right)$$

with  $\tau > 0$ ,  $n \geq 0$  and  $p^0 = 0$ . Thus

$$p_{i,j,k}^{n+1} = \frac{p_{i,j,k}^n + \tau (\nabla(\operatorname{div} p^n - v/\theta))_{i,j,k}}{1 + \tau |(\nabla(\operatorname{div} p^n - v/\theta))_{i,j,k}|}. \quad (3.16)$$

For a detailed proof of convergence, we refer to [10]. The main result is:

**Theorem 3.4** (convergence of the fixed point iteration)

For  $\tau \leq \frac{1}{8}$ , the sequence  $\theta \operatorname{div} p^n$  converges to  $P_{\theta K}(v)$ , the solution of the dual problem (D), as  $n \rightarrow \infty$ .

The solution of the primal problem ( $\operatorname{ROF}_d$ ) can be computed by

$$\bar{u} = v - P_{\theta K}(v). \quad (3.17)$$

In summary, the variational denoising problem ( $\operatorname{ROF}_d$ ) can be solved by Algorithm 1. We implemented Chambolle's algorithm for both two and three-dimensional data. A result for the 2D case is shown in Fig. 3.2. It can be seen that, as desired, the noise is substantially removed, while sharp edges are retained.

---

**Algorithm 1: VariationalDenoising**

---

**input** : image  $v \in X$   
           weight parameter  $\theta > 0$   
           step size  $0 \leq \tau \leq 1/8$   
           stop parameter  $\epsilon > 0$   
**output**: image  $u \in X$

$p = p^{\text{new}} = 0$   
 $\text{dist} = \infty$

**while**  $\text{dist} > \epsilon$  **do**  
     **for**  $i = 1$  **to**  $N$  **do**  
         **for**  $j = 1$  **to**  $M$  **do**  
             **for**  $k = 1$  **to**  $L$  **do**  
                  $p_{i,j,k}^{\text{new}} = \frac{p_{i,j,k} + \tau(\nabla(\text{div } p - v/\theta))_{i,j,k}}{1 + \tau|(\nabla(\text{div } p - v/\theta))_{i,j,k}|}$   
             |  
             |  
             |  
          $\text{dist} = \|p - p^{\text{new}}\|$   
          $p = p^{\text{new}}$   
      $u = v - \theta \text{div } p$

**return**  $u$

---



(a)



(b)

**Figure 3.2:** An example of variational image denoising. (a) Noisy input image. (b) Denoised result obtained by Chambolle's algorithm.

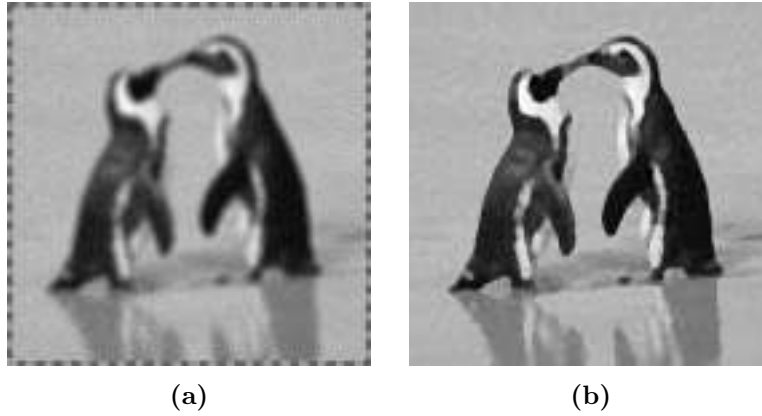
### 3.5 Further Applications of Total Variation Regularization

The variational image denoising model introduced by Rudin, Osher, and Fatami (ROF) is the most established application of total variation in image processing and computer vision. To underline the variety of methods that make use of this special regularization term, we give some more examples here:

- **Deblurring:** The aim of *image deblurring* is to reconstruct a sharp image from a blurred and noisy input image  $v$ ; see Fig. 3.3. Assume that the blur can be modeled by a convolution  $u * k$  with known convolution kernel  $k \in L^1(\Omega)$ . Then a possible restoration model is

$$\min_{u \in BV(\Omega)} J(u) + \lambda \int_{\Omega} |u * k - v|^2 dx.$$

Under certain conditions, the existence and uniqueness of a solution can be shown, see, for example, [7, Beispiel 6.127]. For an introduction to image deblurring, including some physical background of blur, we refer to [14, Chapter 5].



**Figure 3.3:** Variational image deblurring. (a) Blurred image. (b) Deblurred result. The images are obtained from [7].

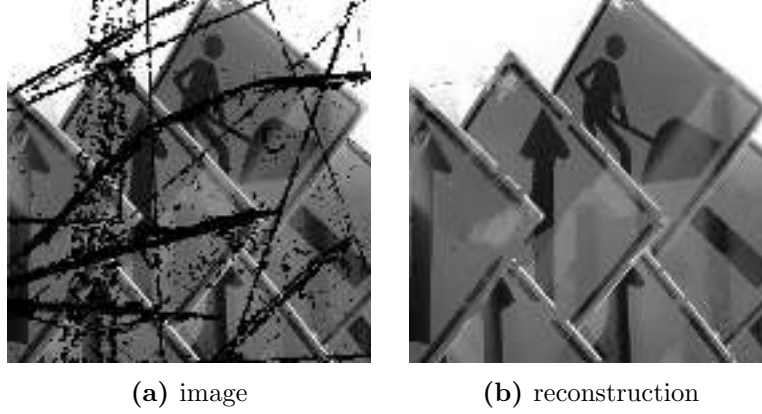
- **Inpainting:** Given a (possibly noisy and/or blurred) image  $v$  on a domain  $\Omega$  with missing data on a subset  $D \subset \Omega$ , *inpainting* describes the reconstruction of the image data on the whole image domain. See Fig. 3.4 for an illustration. A possible variational approach to solve this problem is (see [7, Beispiel 6.128])

$$\min_{u \in BV(\Omega)} J(u) + I_{\{w \in L^2(\Omega) : w|_{\Omega \setminus D} = v|_{\Omega \setminus D}\}}(u),$$

where  $I$  denotes the indicator functional (see Definition 2.12). More about the inpainting problem and its application in different fields, like digital zooming, text removal, and image decoding, can be found in [14, Chapter 6].

- **Optical flow:** Given a sequence of images, *optical flow* methods measure the motion between two consecutive frames  $I_1$  and  $I_2$ . This motion can be represented as flow field  $d : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ . Zach et. al [60] propose the following optimization problem

$$\min_{d \in BV(\Omega)} J(d) + \lambda \int_{\Omega} |I_1(x) - I_2(x + d(x))| dx,$$



**Figure 3.4:** Variational inpainting. (a) Image with missing data. (b) Reconstruction. The images are obtained from [7].

which can be solved using a dual approach.

- **Video restoration:** Similar to image deblurring, *video restoration* aims to recover a sharp video from given degraded video data  $v$ . In [13], Chan et al. represent video signals as three-dimensional functions  $u(x, y, t)$ , where  $t$  denotes the time coordinate. Assuming the video signal has been degraded by an operator  $H$ , one of the following minimization problems

$$\min_{u \in BV(\Omega)} J(u) + \lambda \int_{\Omega} (Hu - v)^2 dx,$$

$$\min_{u \in BV(\Omega)} J(u) + \lambda \int_{\Omega} |Hu - v| dx$$

can be used for reconstruction.

Even though used in different applications, total variation regularization often yields very similar optimization schemes. In the following chapter, we present a variational approach for 3D multi-view reconstruction that was proposed by Zach et al. [61]. We will notice that it is connected to the variational denoising problem (ROF) and very similar to the applications mentioned above.

---

## 3D Reconstruction by a Robust Variational Merging of Depth Maps

---

In 2007, Zach et al. [61] suggested a variational approach to reconstruct 3D surfaces from depth maps. Using the depth information from  $K$  points of view, three-dimensional distance fields  $f^m : \Omega \rightarrow [-1, 1]$ ,  $m = 1 \dots K$ ,  $\Omega \subset \mathbb{R}^3$  are generated. Those measure the signed distance to the hypothetical surface. A point  $x$  is supposed to lie in front of the surface if  $0 \leq f^m(x) \leq 1$ , on the surface if  $f^m(x) = 0$ , and behind the surface if  $-1 \leq f^m(x) \leq 0$ . To find an optimal distance field  $u$  that approximates all  $f^m$  simultaneously, the optimization problem

$$\min_{u \in BV(\Omega)} J(u) + \lambda \int_{\Omega} \sum_{m \in I(x)} |u(x) - f^m(x)| dx \quad (Z)$$

is solved approximately, where  $I(x)$  denotes the set of relevant indices of distance fields for point  $x$ . The relevance of a  $f^m$  is linked to the visibility of the point  $x$ , as will be explained in detail in Section 4.2. The reconstructed surface can be extracted from the zero set of the optimal distance field  $u$ .

In the following, we will first analyze the structure of the energy model in (Z) in detail. After discussing the generation of distance fields from depth maps, we will then present the algorithm proposed by Zach et al. and prove its convergence to an optimal solution.

### 4.1 TV- $L^1$ Energy Model

We define the energy functional

$$E(u) := J(u) + \lambda \int_{\Omega} \sum_{m \in I(x)} |u(x) - f^m(x)| dx. \quad (4.1)$$

Before presenting an algorithm to find an approximate minimizer of  $E(u)$  we will motivate the choice of the data fidelity term  $\int_{\Omega} \sum_{m \in I(x)} |u(x) - f^m(x)|$  and the regularization term  $J(u)$ , respectively.

- **$L^1$ -data fidelity:** As mentioned above, an optimal distance field  $u$  should approximate all input fields  $f^m$  at the same time. This means the distance  $\|u - f^m\|$  must be minimized

for all  $m = 1, \dots, K$ . But which norm  $\|\cdot\|$  is appropriate for measuring this distance?

Since the distance fields  $f^m$  usually are degraded because of noise, occlusions, and erroneous matchings during the depth map construction, the norm of choice should be sufficiently robust. In [42], Nikolova showed that using the  $L^1$ -norm results in an implicit and stable outlier detection. Combining  $L^1$ -data fidelity with a convex edge preserving regularization, uncorrupted data can be kept, while outliers are replaced by regularized values. Moreover, the convexity of the norm yields convexity of  $\int_{\Omega} \sum_{m \in I(x)} |u(x) - f^m(x)|$ .

- **TV-Regularization:** A reconstructed surface should be smooth, but still represent the sharp features of an object. As mentioned in Chapter 3, the total variation functional  $J$  embodies an appropriate regularization term to deal with the tradeoff between smoothness and preservation of edges. Furthermore,  $J(u)$  is closely linked to the perimeter of level sets of  $u \in BV(\Omega)$  as stated by Federer and Fleming in the so-called *co-area* formula (see [11]):

$$J(u) = \int_{-\infty}^{\infty} \text{Per}(\{u > s\}; \Omega) ds$$

where  $\text{Per}(S; \Omega)$  denotes the perimeter of a measurable set  $S \subset \Omega$  in  $\Omega$ . This means that using total variation regularization penalizes the perimeter of level sets of  $u$ . For the 3D reconstruction problem, this is equivalent to penalizing the surface area, and consequently yields minimal surfaces. Finally, as shown in Theorem 3.2 (ii), the total variation functional  $J$  is convex.

Consequently, the energy functional  $E$  is convex because it is the sum of convex functionals and presents a reasonable modeling of the multi-view stereo reconstruction problem. Note, however, that  $E$  is not strictly convex. Therefore, uniqueness of a solution cannot be guaranteed. Moreover,  $E$  is not differentiable because of the  $L^1$  term.

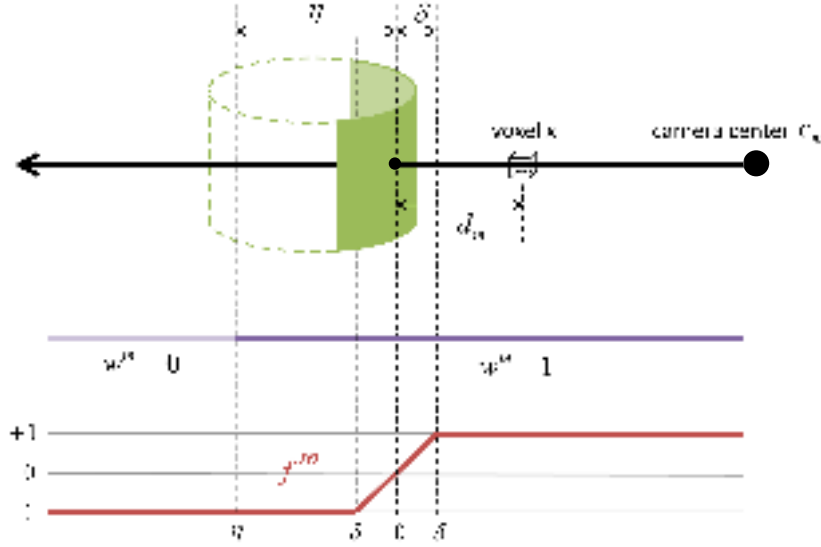
## 4.2 Conversion of Depth Maps into Distance Fields

Suppose that, for every camera position, a corresponding depth map  $r_m : D_m \rightarrow \mathbb{R}$  on the image domain  $D_m$ ,  $m = 1, \dots, K$  is given. For every pixel  $p \in D_m$ , the value  $r_m(p)$  describes the distance from a point on the object surface to the camera center  $c_m$ . Depending on the quality of the method used for generating depth maps from input images, there is more or less noise in the depth maps that must be considered.

We assume a voxel volume that completely contains the hypothetical object surface determined from the depth information  $r_m$ . For every voxel  $x$ , the signed distance  $d_m(x)$  to the object surface along the line of sight is measured. Therefore, a line through the voxel center and the camera center  $c_m$  is created and the intersection with the hypothetical surface is determined. This procedure is illustrated in Fig. 4.1. If there is no intersection, no distance value can be assigned. To also handle these cases, Zach et al. introduce a binary confidence measure  $w^m(x) \in \{0, 1\}$ , with  $w^m(x) = 0$  if no distance value exists. The distance field  $f^m$  for voxels with valid distance  $d_m$  is defined as

$$f^m(x) = \begin{cases} -1 & \text{if } d_m(x) < -\delta \\ \frac{1}{\delta} \cdot d_m(x) & \text{if } -\delta \leq d_m(x) \leq \delta \\ 1 & \text{if } d_m(x) > \delta \end{cases}$$

where  $\delta > 0$  defines the width of the region around the surface that is considered. The bigger  $\delta$ , the more deviations in the depth maps, i.e., noise, is taken into account. Hence, voxels with



**Figure 4.1:** Generation of distance fields from depth maps.

$d_m(x) > \delta$  are assumed to surely lie in front of the object and receive a value of 1. Analogously, the points with  $d_m(x) < -\delta$  are considered to lie behind the surface and receive a value of -1. Voxels in between are potential surface points and their distance is truncated to values in  $[-1, 1]$ . Since points behind the surface are hidden in the respective view, one does not know whether they lie inside the object, on the object surface on the back side, or behind the object. To deal with this, Zach et al. introduce an additional parameter  $\eta > \delta$  that controls the range behind the hypothetical surface that is supposed to lie within the object. Voxels with  $d_m(x) < -\eta$ , i.e., outside this region, should not be considered and therefore receive confidence zero:  $w^m(x) = 0$ .

For the surface reconstruction, we only take voxels into account that have confidence one, that is, voxels  $x$  with  $w^m(x) = 1$ . For every voxel, we define the set of relevant indices as

$$I(x) = \{m \in \{1, \dots, K\} : w^m(x) = 1\}.$$

## 4.3 Optimization Scheme

Since the optimization problem (Z) is difficult to solve directly, Zach et al. introduce an approximation  $E_\theta$  of the energy functional  $E$  (4.1) and provide an optimization scheme to obtain a minimizer of  $E_\theta$ . We will present their approach and show that the algorithm is convergent and yields a solution that is close to the solution of (Z).

### 4.3.1 Discrete Optimization Problem

Because we deal with distance fields defined on a discrete voxel grid, from now on we will consider the discrete case: Let  $X$  be the Euclidean space  $\mathbb{R}^{N \times M \times L}$  with the inner product  $(u, v)_X$  and the norm  $\|u\|_X$  as defined in Chapter 3.3.2. Moreover, we introduce the product space  $Y := X \times X$  with

$$\|(u, v)\|_Y = \sqrt{\|u\|_X^2 + \|v\|_X^2}.$$

For each voxel  $(i, j, k)$ , the discrete confidence measure for distance field  $f^m \in X$  is denoted by  $w_{i,j,k}^m$  and the set of relevant indices can be written as

$$I_{i,j,k} = \{m \in \{1, \dots, K\} : w_{i,j,k}^m = 1\}.$$

For the sake of simplicity, from now on we will write

$$\sum_{i,j,k} \text{ for } \sum_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M \\ 1 \leq k \leq L}} \text{ and } \sum_m \text{ for } \sum_{m \in I_{i,j,k}}.$$

Then, for  $u \in X$ ,  $f^m \in X$ ,  $m = 1, \dots, K$  the energy functional in (4.1) becomes

$$E(u) = J(u) + \lambda \sum_{i,j,k} \sum_m |u_{i,j,k} - f_{i,j,k}^m|,$$

where  $J$  is the discrete total variation introduced in Chapter 3.3.2. We define the set of input distance fields as  $F := \{f^m : m \in \{1, \dots, K\}\}$  and

$$G_F(u) := \sum_{i,j,k} \sum_m |u_{i,j,k} - f_{i,j,k}^m|.$$

Then,  $E(u) = J(u) + \lambda G_F(u)$  and the discrete optimization problem reads

$$\min_{u \in X} J(u) + \lambda G_F(u). \quad (\text{P})$$

### 4.3.2 Approximation of the Energy Functional

Since  $E$  is not continuous differentiable, (P) is a non-smooth optimization problem that is difficult to solve. Therefore, Zach et al. consider the approximation  $E_\theta : X \times X \rightarrow \mathbb{R}$

$$E_\theta(u, v) = J(u) + \frac{1}{2\theta} \|u - v\|_X^2 + \lambda G_F(v),$$

where  $\theta > 0$  is a small constant such that  $v$  provides a close approximation of  $u$ . Instead of solving (P), they derive an optimization scheme to find the unique minimizer of  $E_\theta$ . The corresponding minimization problem reads

$$\min_{(u,v) \in X \times X} J(u) + \frac{1}{2\theta} \|u - v\|_X^2 + \lambda G_F(v). \quad (\text{P}_\theta)$$

Without giving a proof, Zach et al. claim that solving  $(\text{P}_\theta)$  yields an approximate solution of (P) that can be obtained by a globally convergent optimization scheme.

In the following, we will show that for  $\theta \rightarrow 0$  the solution of  $(\text{P}_\theta)$  converges to a solution of (P). Moreover, we will present the algorithm suggested by Zach et al. and prove its convergence to the optimal solution of  $(\text{P}_\theta)$ .



As sum of continuous and convex functionals  $E_\theta$  is continuous and convex on  $X \times X$ . In particular,  $E_\theta$  is lower semi-continuous. Moreover,

$$\begin{aligned}
 E_\theta(u, v) &= \underbrace{J(u)}_{\geq 0} + \frac{1}{2\theta} \|u - v\|_X^2 + \underbrace{\lambda G_F(v)}_{\geq 0} \\
 &\geq \frac{1}{2\theta} \|u - v\|_X^2 \\
 &= \frac{1}{2\theta} (u - v, u - v)_X \\
 &= \frac{1}{2\theta} [(u, u)_X - 2(u, v)_X + (v, v)_X] \\
 &= \frac{1}{2\theta} [\|u\|_X^2 + \|v\|_X^2 - 2(u, v)_X] \\
 &= \frac{1}{2\theta} [\|(u, v)\|_Y^2 - 2(u, v)_X].
 \end{aligned}$$

Consequently, if  $\|(u, v)\|_Y \rightarrow \infty$  then  $E_\theta \rightarrow \infty$ , which means  $E_\theta$  is coercive.

Using convexity, lower semi-continuity, and coercivity of  $E_\theta$  in analogy to [3], [4], it can be shown that problem  $(P_\theta)$  has a unique solution in  $X \times X$ .

**Theorem 4.1** (unique solution of  $(P_\theta)$ )

*There exists a unique solution  $(\bar{u}, \bar{v}) \in X \times X$  of problem  $(P_\theta)$ .*

Now, similar to [4], we show the relationship between the optimization problems (P) and  $(P_\theta)$ . Let  $\theta_n$  be a decreasing sequence with  $\theta_n > 0$  and  $\theta_n \rightarrow 0$  as  $n \rightarrow \infty$ . Furthermore, denote by  $(u_{\theta_n}, v_{\theta_n})$  the minimizer of  $E_{\theta_n}$ .

**Theorem 4.2** (solution of (P))

*The sequence  $(u_{\theta_n}, v_{\theta_n})$  is bounded and for any cluster point  $(u^0, v^0) \in X \times X$ ,  $u^0$  is a solution of (P).*

**Proof:**  $(u_{\theta_n}, v_{\theta_n})$  minimizes  $E_{\theta_n}$ . Hence,

$$E_{\theta_n}(u_{\theta_n}, v_{\theta_n}) \leq E_{\theta_n}(u, v) \quad \forall (u, v) \in X \times X$$

and particularly

$$E_{\theta_n}(u_{\theta_n}, v_{\theta_n}) \leq E_{\theta_n}(0, 0) = \lambda \sum_{i,j,k} \sum_m |f_{i,j,k}^m|, \quad (4.2)$$

which means  $E_{\theta_n}(u_{\theta_n}, v_{\theta_n})$  is bounded. Since  $E_{\theta_n}$  is coercive, it follows that the sequence  $(u_{\theta_n}, v_{\theta_n})$  is bounded in  $X \times X$ . Consequently, there exists a convergent subsequence. Hence, up to an extraction there is a  $(u^0, v^0) \in X \times X$  with  $(u_{\theta_n}, v_{\theta_n}) \rightarrow (u^0, v^0)$ . It remains to show that  $u^0$  solves (P). Multiplying with  $\theta_n > 0$  in (4.2) yields

$$\theta_n J(u_{\theta_n}) + \frac{1}{2} \|u_{\theta_n} - v_{\theta_n}\|_X^2 + \theta_n G_F(v_{\theta_n}) \leq \theta_n \lambda \sum_{i,j,k} \sum_m |f_{i,j,k}^m|$$

and passing to the limit  $\theta_n \rightarrow 0$  we obtain

$$\frac{1}{2} \|u^0 - v^0\|_X^2 \leq 0.$$

Thus,  $u^0 = v^0$ . Then, for all  $u \in X$

$$\begin{aligned} E(u) &= J(u) + \lambda G_F(u) \\ &= J(u) + \underbrace{\frac{1}{2\theta_n} \|u - u\|_X^2}_{=0} + \lambda G_F(u) \\ &= E_{\theta_n}(u, u). \end{aligned}$$

Using the optimality of  $(u_{\theta_n}, v_{\theta_n})$ , we further obtain

$$\begin{aligned} E_{\theta_n}(u, u) &\geq E_{\theta_n}(u_{\theta_n}, v_{\theta_n}) \\ &= J(u_{\theta_n}) + \underbrace{\frac{1}{2\theta_n} \|u_{\theta_n} - v_{\theta_n}\|_X^2}_{\geq 0} + \lambda G_F(v_{\theta_n}) \\ &\geq J(u_{\theta_n}) + \lambda G_F(v_{\theta_n}). \end{aligned}$$

Hence,

$$E(u) \geq J(u_{\theta_n}) + \lambda G_F(v_{\theta_n}) \quad \forall u \in X.$$

Passing to the limit  $\theta_n \rightarrow 0$  yields

$$E(u) \geq J(u^0) + \lambda G_F(v^0) \quad \forall u \in X$$

and using  $u^0 = v^0$  results in

$$E(u) \geq J(u^0) + \lambda G_F(u^0) = E(u^0) \quad \forall u \in X.$$

Thus,  $u^0$  is a solution of (P). □

As a consequence, solving  $(P_\theta)$  for a fixed  $\theta > 0$  yields a solution  $(u_\theta, v_\theta)$  where  $u_\theta$  is close to a solution of (P) if  $\theta$  is sufficiently small.

### 4.3.3 Alternating Descent Approach

Since  $E_\theta(u, v)$  is convex, the solution can be found by alternately minimizing with respect to  $u$  and  $v$  separately: Zach et al. propose the following iterative optimization scheme:

- For  $v$  being fixed, minimize  $E_\theta$  with respect to  $u$ , that is, solve

$$\min_{u \in X} J(u) + \frac{1}{2\theta} \|u - v\|_X^2. \quad (\text{P1})$$

This is the variational denoising problem (ROF<sub>d</sub>) introduced in Chapter 3.3.2 and its unique solution is given by  $\bar{u} = v - P_{\theta K}(v)$  (3.17).

- For  $u$  being fixed, minimize  $E_\theta$  with respect to  $v$ , that is, solve

$$\min_{v \in X} \frac{1}{2\theta} \|u - v\|_X^2 + \lambda G_F(v). \quad (\text{P2})$$

As will be shown in Section 4.3.5, this optimization has as unique solution  $\bar{v} \in X$  that can be obtained by a point-wise algorithm.

Moreover, Zach et al. state that solving (P1) and (P2) by turns is globally convergent and results in a minimizer of  $E_\theta$ . In what follows, we will give a detailed proof of this assertion, which is derived in analogy to [4].

Let  $u_n$  and  $v_n$  denote the solutions of (P1) and (P2) in the  $n$ th iteration, respectively.

**Theorem 4.3** (convergence of the alternating optimization scheme)

*The sequence  $(u_n, v_n)$  converges to the minimizer of  $E_\theta$ .*

**Proof:** By definition,  $E_\theta(u_n, v_n) \geq 0$  and since  $u_n$  and  $v_n$  minimize  $E_\theta$  with respect to  $u$  and  $v$ , respectively, we obtain

$$E_\theta(u_n, v_n) \geq E_\theta(u_{n+1}, v_n) \geq E_\theta(u_{n+1}, v_{n+1}). \quad (4.3)$$

Hence, the sequence  $E_\theta(u_n, v_n)$  is non-increasing and bounded and therefore converges to an  $\bar{E}_\theta \in \mathbb{R}$  as  $n \rightarrow \infty$ . We will show that  $E_\theta(u_n, v_n)$  converges to the minimum of  $E_\theta$ , that is,

$$\bar{E}_\theta = \min_{(u,v) \in X \times X} E_\theta(u, v).$$

As shown above,  $E_\theta$  is coercive and since  $E_\theta(u_n, v_n)$  converges, the sequence  $(u_n, v_n)$  is bounded in  $X \times X$ . Consequently, there exists a convergent subsequence  $(u_{n_k}, v_{n_k})$  with

$$(u_{n_k}, v_{n_k}) \rightarrow (\hat{u}, \hat{v}) \in X \times X.$$

Furthermore, in iteration  $n_k \in \mathbb{N}$  the functional  $E_\theta(u_{n_k}, v)$  is minimized by  $v_{n_k}$  and in the next iteration  $n_k + 1$  the functional  $E_\theta(u, v_{n_k})$  is minimized by  $u_{n_k+1}$ . Thus, we have for all  $n_k \in \mathbb{N}$ :

$$E_\theta(u_{n_k}, v_{n_k}) \leq E_\theta(u_{n_k}, v) \quad \forall v \in X \quad (4.4)$$

$$E_\theta(u_{n_k+1}, v_{n_k}) \leq E_\theta(u, v_{n_k}) \quad \forall u \in X. \quad (4.5)$$

We know that  $u_{n_k} \rightarrow \hat{u}$  as  $n_k \rightarrow \infty$ . Now, we will show that also the sequence  $u_{n_k+1}$  converges to  $\hat{u}$ .

Let  $\tilde{u}$  be a cluster point of  $u_{n_k+1}$ . Since the sequence  $E_\theta(u_n, v_n)$  converges to  $\bar{E}_\theta$  every subsequence converges to  $\bar{E}_\theta$ . In particular,  $E_\theta(u_{n_k}, v_{n_k}) \rightarrow \bar{E}_\theta$  and  $E_\theta(u_{n_k+1}, v_{n_k+1}) \rightarrow \bar{E}_\theta$  as  $n_k \rightarrow \infty$ . Using (4.3), we obtain

$$E_\theta(u_{n_k}, v_{n_k}) \geq E_\theta(u_{n_k+1}, v_{n_k}) \geq E_\theta(u_{n_k+1}, v_{n_k+1})$$

and passing to the limit  $n_k \rightarrow \infty$  yields

$$\bar{E}_\theta \geq E_\theta(\tilde{u}, \hat{v}) \geq \bar{E}_\theta$$

since  $E_\theta$  is continuous. Hence,  $E_\theta(\tilde{u}, \hat{v}) = \bar{E}_\theta$  and using continuity of  $E_\theta$  it follows

$$\bar{E}_\theta = E_\theta(\tilde{u}, \hat{v}) = E_\theta(\hat{u}, \hat{v}).$$

This means

$$J(\tilde{u}) + \frac{1}{2\theta_n} \|\tilde{u} - \hat{v}\|_X^2 + \lambda G_F(\hat{v}) = J(\hat{u}) + \frac{1}{2\theta_n} \|\hat{u} - \hat{v}\|_X^2 + \lambda G_F(\hat{v}).$$

Thus,

$$J(\tilde{u}) + \frac{1}{2\theta_n} \|\tilde{u} - \hat{v}\|_X^2 = J(\hat{u}) + \frac{1}{2\theta_n} \|\hat{u} - \hat{v}\|_X^2. \quad (4.6)$$

As shown in (3.17),

$$u_{n_k+1} = v_{n_k} - P_{\theta K}(v_{n_k})$$

and for  $n_k \rightarrow \infty$  we obtain

$$\tilde{u} = \hat{v} - P_{\theta K}(\hat{v}).$$

Hence,  $\tilde{u}$  solves (P1) for  $v = \hat{v}$ . Using (4.6), by uniqueness of the solution of (P1), we deduce  $\tilde{u} = \hat{u}$ . Passing to the limit  $n_k \rightarrow \infty$  in (4.4) and (4.5) we further obtain

$$\begin{aligned} E_{\theta}(\hat{u}, \hat{v}) &\leq E_{\theta}(\hat{u}, v) \quad \forall v \in X \\ E_{\theta}(\hat{u}, \hat{v}) &\leq E_{\theta}(u, \hat{v}) \quad \forall u \in X. \end{aligned}$$

That is,

$$\begin{aligned} E_{\theta}(\hat{u}, \hat{v}) &= \min_{v \in X} E_{\theta}(\hat{u}, v) \\ E_{\theta}(\hat{u}, \hat{v}) &= \min_{u \in X} E_{\theta}(u, \hat{v}), \end{aligned}$$

which is equivalent to

$$\begin{aligned} \hat{v} \text{ solves } & \min_{v \in X} \frac{1}{2\theta} \|\hat{u} - v\|_X^2 + \lambda G_F(v) \\ \hat{u} \text{ solves } & \min_{u \in X} J(u) + \frac{1}{2\theta} \|u - \hat{v}\|_X^2. \end{aligned}$$

Using the optimality condition for convex minimization problems in Theorem 2.7, it follows that

$$0 \in \partial \left( \frac{1}{2\theta} \|\hat{u} - \hat{v}\|_X^2 + \lambda G_F(\hat{v}) \right) = \frac{1}{\theta}(\hat{u} - \hat{v}) + \lambda \partial G_F(\hat{v}) \quad (4.7)$$

$$0 \in \partial \left( J(\hat{u}) + \frac{1}{2\theta} \|\hat{u} - \hat{v}\|_X^2 \right) = \partial J(\hat{u}) + \frac{1}{\theta}(\hat{u} - \hat{v}). \quad (4.8)$$

The subdifferential of  $E_{\theta}$  is given by

$$\partial E_{\theta}(u, v) = \left( \begin{array}{c} \partial J(u) + \frac{1}{\theta}(u - v) \\ \frac{1}{\theta}(u - v) + \lambda \partial G_F(v) \end{array} \right)$$

and with (4.7) and (4.8) we obtain

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \partial E_{\theta}(\hat{u}, \hat{v}).$$

Using Theorem 2.7 again, we conclude that  $(\hat{u}, \hat{v})$  minimizes  $E_{\theta}$ . Altogether, we have

$$\lim_{n \rightarrow \infty} E_{\theta}(u_n, v_n) = \overline{E}_{\theta} = E_{\theta}(\hat{u}, \hat{v}) = \min_{(u,v) \in X \times X} E_{\theta}(u, v).$$

Since, according to Theorem 4.1, the solution of  $(P_{\theta})$  is unique, the whole sequence  $(u_n, v_n)$  converges to  $(\hat{u}, \hat{v})$  as  $n \rightarrow \infty$ . Hence, alternating minimization of  $E_{\theta}$  with respect to  $u$  and  $v$  converges to the minimizer  $(\hat{u}, \hat{v})$  of  $E_{\theta}$ .  $\square$

In the following, we will present algorithms to solve the optimization problems (P1) and (P2) and complete this chapter with the whole reconstruction algorithm.

#### 4.3.4 Denoising Step

As shown in Chapter 3.4.3, the variational denoising problem

$$\min_{u \in X} J(u) + \frac{1}{2\theta} \|u - v\|_X^2 \quad (\text{P1})$$

can be solved using the iterative fixed point scheme (3.16) introduced by Chambolle. For the reader's convenience, we recall the corresponding Algorithm 1. Note that the only modification that has been performed is the substitution of the word “image” by “distance field”.

---

**Algorithm 1: VariationalDenoising**


---

**input** : distance field  $v \in X$   
 weight parameter  $\theta > 0$   
 step size  $0 \leq \tau \leq 1/8$   
 stop parameter  $\epsilon > 0$   
**output**: distance field  $u \in X$

$p = p^{\text{new}} = 0$   
 $\text{dist} = \infty$

**while**  $\text{dist} > \epsilon$  **do**  
     **for**  $i = 1$  **to**  $N$  **do**  
         **for**  $j = 1$  **to**  $M$  **do**  
             **for**  $k = 1$  **to**  $L$  **do**  
                  $p_{i,j,k}^{\text{new}} = \frac{p_{i,j,k} + \tau(\nabla(\text{div } p - v/\theta))_{i,j,k}}{1 + \tau|(\nabla(\text{div } p - v/\theta))_{i,j,k}|}$   
              $\text{dist} = \|p - p^{\text{new}}\|$   
          $p = p^{\text{new}}$   
      $u = v - \theta \text{div } p$

**return**  $u$

---

#### 4.3.5 Generalized Thresholding Step

For solving the second optimization problem

$$\min_{v \in X} \frac{1}{2\theta} \|u - v\|_X^2 + \lambda G_F(v), \quad (\text{P2})$$

Zach et. al [61] suggest a point-wise optimization algorithm: Define

$$E_u^2(v) = \frac{1}{2\theta} \|u - v\|_X^2 + \lambda G_F(v).$$

Then,

$$\begin{aligned} E_u^2(v) &= \frac{1}{2\theta} \sum_{i,j,k} (u_{i,j,k} - v_{i,j,k})^2 + \lambda \sum_{i,j,k} \sum_m |v_{i,j,k} - f_{i,j,k}^m| \\ &= \sum_{i,j,k} \left[ \frac{1}{2\theta} (u_{i,j,k} - v_{i,j,k})^2 + \lambda \sum_m |v_{i,j,k} - f_{i,j,k}^m| \right]. \end{aligned}$$

As sum of non-negative terms  $E_u^2(v)$  is minimal if all summands are minimal. This results in a one-dimensional minimization problem

$$\min_{v_{i,j,k} \in \mathbb{R}} \frac{1}{2\theta} (u_{i,j,k} - v_{i,j,k})^2 + \lambda \sum_m |v_{i,j,k} - f_{i,j,k}^m|$$

for each voxel  $(i, j, k)$ . These problems can be solved independently from each other. For convenience, let us fix  $i, j$ , and  $k$  and write  $u = u_{i,j,k}$ ,  $f^m = f_{i,j,k}^m$  and  $I = I_{i,j,k}$ . Then, the optimization problem for voxel  $(i, j, k)$  reads

$$\min_{v \in \mathbb{R}} \frac{1}{2\theta} (u - v)^2 + \lambda \sum_{m \in I} |v - f^m|. \quad (4.9)$$

Since

$$\sum_{m \in I} |v - f^m|$$

does not depend on the order of the  $f^m$ , we can assume  $f^m \leq f^{m+1}$  with  $f^0 := -\infty$  and  $f^{|I|+1} := \infty$ . In the interior of  $(f^l, f^{l+1})$ , for  $l \in \{0, \dots, |I|\}$

$$\frac{1}{2\theta} (u - v)^2 + \lambda \sum_{i \in I} |v - f^m| \quad (4.10)$$

is strictly convex and differentiable with respect to  $v$ . Consequently, the minimum is unique. If there is a  $v_1 \in (f^l, f^{l+1})$  that minimizes (4.10) for some  $l \in \{0, \dots, |I|\}$ , it must be a stationary point. Hence,

$$\frac{1}{\theta} (v_1 - u) + \lambda \sum_{m \in I(x)} \text{sgn}(v_1 - f^m) = 0.$$

Moreover,  $f^l < v_1 < f^{l+1}$  and since the  $f^m$  are sorted

$$\sum_{m \in I} \text{sgn}(v_1 - f^m) = \# \{m : v_1 - f^m > 0\} - \# \{m : v_1 - f^m < 0\} = l - (|I| - l) = 2l - |I|.$$

This means, if there is a

$$v_1 = u - \theta \lambda (2l - |I|)$$

that lies within  $(f^l, f^{l+1})$  for some  $l \in \{0, \dots, |I|\}$ , then  $v = v_1$  is the optimal solution. If there is no such  $v_1$ , the solution must lie on the boundary of one of these intervals and therefore must be among the  $f^m$ , i.e.,

$$v = \arg \min_{v_2 \in \{f^l : l \in \{0, \dots, |I|\}\}} \left( \frac{1}{2\theta} (u - v_2)^2 + \lambda \sum_{m \in I} |v_2 - f^m| \right).$$

All in all, (P2) can be solved by Algorithm 2.

## 4.4 Reconstruction Algorithm

Summing up, the whole multi-view stereo reconstruction algorithm proposed by Zach et al. is described in Algorithm 3.

**Algorithm 2: GeneralizedThresholding**


---

**input** : distance field  $u$   
           sorted distance fields  $f^m, m = 1, \dots, K$   
           index sets  $I_{i,j,k}, i = 1, \dots, N, j = 1, \dots, M, k = 1, \dots, L$   
           weight parameters  $\theta, \lambda > 0$

**output**: distance field  $v$

**for**  $i = 1$  **to**  $N$  **do**  
   **for**  $j = 1$  **to**  $M$  **do**  
     **for**  $k = 1$  **to**  $L$  **do**  
        $f_{i,j,k}^0 = -\infty$   
        $f_{i,j,k}^{|I_{i,j,k}|+1} = \infty$   
       **for**  $l = 0$  **to**  $|I_{i,j,k}|$  **do**  
          $v_1 = u_{i,j,k} - \theta\lambda(2l - |I_{i,j,k}|)$   
         **if**  $f^l < v_1 < f^{l+1}$  **then**  
            $v_{i,j,k} = v_1$   
           **break**  
       **if** no value has been assigned to  $v_{i,j,k}$  **then**  
         
$$v_{i,j,k} = \arg \min_{v_2 \in \{f_{i,j,k}^m\}} \left( \frac{1}{2\theta} (u_{i,j,k} - v_2)^2 + \lambda \sum_{m \in I_{i,j,k}} |v_2 - f_{i,j,k}^m| \right)$$
  
     **return**  $v$

---

**Algorithm 3: SurfaceReconstruction**


---

**input** : depth maps  $r_m, m = 1, \dots, K$   
           distance parameters  $\eta > \delta > 0$   
           weight parameters  $\theta, \lambda > 0$   
           optimization parameters  $0 \leq \tau \leq 1/8, \epsilon_1 > 0, \epsilon_2 > 0$

**output**: surface  $S$

**for**  $m = 1$  **to**  $K$  **do**  
    $f^m = \text{CreateDistanceField}(r_m, \delta, \eta)$   
 $F = [f^1, f^2, \dots, f^m]$   
 Sort ( $F$ )  
 $u = v = 0$

**while**  $\text{dist} > \epsilon_1$  **do**  
    $v_{\text{new}} = \text{GeneralizedThresholding}(u, F, \theta, \lambda)$   
    $u_{\text{new}} = \text{VariationalDenoising}(v_{\text{new}}, \theta, \tau, \epsilon_2)$   
    $\text{dist} = \max(\|u - u_{\text{new}}\|, \|v - v_{\text{new}}\|)$   
    $u = u_{\text{new}}$   
    $v = v_{\text{new}}$

$S = \text{ExtractSurfaceFromDistanceField}(u)$

**return**  $S$

---





## 5.1 Implementation Details

In order to achieve a detailed surface reconstruction, the resolution of the underlying voxel grid must be sufficiently high. Consequently, the construction of distance fields (see Chapter 4.2) results in large data volumes that must be stored. We will introduce two data compression methods that can be used to reduce the memory footprint. Furthermore, we will show how the ascending order of distance field values can be used for an efficient implementation of the generalized thresholding step.

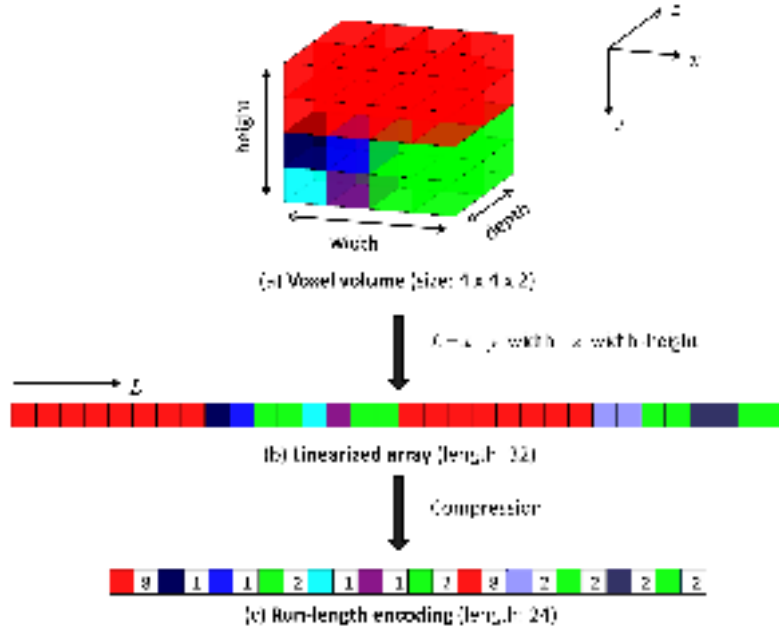
### 5.1.1 Data Compression

The distance fields introduced in Chapter 4.2 are volumes of size  $N \times M \times L$ , where  $N$ ,  $M$ , and  $L$  determine the resolution of the voxel grid. We created 46 distance fields of size  $205 \times 321 \times 151$  for the Middlebury “TempleRing” data set, which required approximately 11.5 GB. Thus, if a direct volumetric representation of the distance fields is used, the resolution (and therefore quality) of the reconstruction is limited.

As described in Chapter 4.2, only voxels within a small region around the hypothetical surface receive values in  $(-1, 1)$ . The majority of voxels, however, is classified as lying in front or behind the surface and therefore receives a value of 1 or  $-1$ , respectively. This can be utilized to compress the distance volumes and thereby reduce the memory footprint. We will present two possible compression methods, namely *run-length encoding* and *octrees*.

#### Run-length Encoding

The concept of run-length encoding is illustrated in Fig. 5.1. In Fig. 5.1a, a small voxel grid is shown, where the color of each voxel represents a certain value. In terms of distance fields, red and green correspond to values of  $-1$  and  $1$ , respectively, while the shades of blue represent values in  $(-1, 1)$ . The voxel volume can be converted into a linearized array, as shown in Fig. 5.1b. A sequence of entries with the same value is called a *run* and can be stored by its value and its length (run-length encoding). Hence, for data with many long runs, the amount of memory can be reduced substantially.



**Figure 5.1:** An example of run-length encoding. (a) Volumetric representation of a small voxel grid. The color of each voxel describes its value. (b) Conversion to a linearized array. (c) Run-length-encoded representation. Sequences of entries with the same value (color) are stored by their value and length.

## Octrees

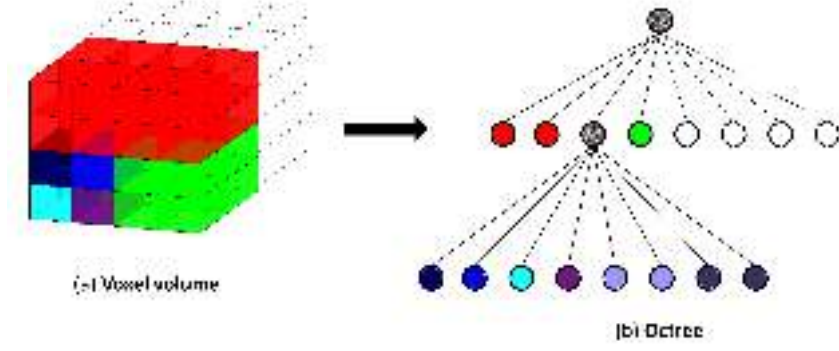
Owing to their characteristic structure, distance fields can also be stored efficiently by means of octrees. An octree is a tree structure that is particularly suitable for compressing data in a three-dimensional space. The space is recursively partitioned into eight cubes until each cube contains one single value only. For distance fields, the octree is built as follows: The root node represents the entire voxel grid. Recursively, the eight children of each node correspond to the eight sub-cubes of the parent voxel cube. If a cube contains only voxels with uniform distance value, it is no longer subdivided; see Fig 5.2. Consequently, each leaf in the tree structure represents a set of neighboring voxels with equal distances to the hypothetical surface.

Octrees are often used in computer graphics and also seem promising for the compression of the distance fields defined in Chapter 4.2. Because of its simple implementation, however, we chose run-length encoding for our implementation of the surface reconstruction algorithm. In this way, the size of the 46 distance fields for the Middlebury “TempleRing” data set could be reduced from 11.5 GB to 2 GB.

### 5.1.2 Efficient Implementation of the Generalized Thresholding Step

Recall the generalized thresholding step in Algorithm 2. Again, let us fix  $i$ ,  $j$ , and  $k$  and set  $v = v_{i,j,k}$ ,  $u = u_{i,j,k}$ ,  $f^m = f_{i,j,k}^m$ , and  $I = I_{i,j,k}$ . In the worst case, there is no

$$v_1 = u - \theta\lambda(2l - |I|)$$



**Figure 5.2:** An example of an octree. (a) Volumetric representation of parts of a voxel grid. The color of each voxel describes its value. (b) Representation in an octree structure. The voxel grid is subdivided in cubes with uniform voxel values (colors).

that lies within  $(f^l, f^{l+1})$  for some  $l \in \{0, \dots, |I|\}$ . Thus,

$$\frac{1}{2\theta}(u - f^l)^2 + \lambda \sum_{m \in I} |f^l - f^m| \quad (5.1)$$

must be computed for all  $l = 0, \dots, |I|$  to obtain

$$v = \arg \min_{v_2 \in \{f^l\}} \left( \frac{1}{2\theta}(u - v_2)^2 + \lambda \sum_{m \in I} |v_2 - f^m| \right).$$

These computations can be very costly, especially for a high number of input distance fields. According to Zach et al. [61], it is, however, possible to restrict the search for  $v$  to a certain interval, as described next.

### Restriction of the Search Space

Let  $\bar{f} \in \mathbb{R}$  denote the median of the set  $\{f^m : m \in I\}$ . It is known (see [26], Appendix 6.8) that the median of a finite set  $\{a_j : j \in J\}$  of real numbers minimizes the cost function  $C(x) = \sum_j |x - a_j|$ . Hence,  $v = \bar{f}$  is the minimizer of  $\sum_{i \in I} |v - f^m|$ . Furthermore,  $\frac{1}{2\theta}(u - v)^2$  is minimal for  $v = u$ . Therefore, the solution of the optimization problem

$$\min_{v \in \mathbb{R}} \frac{1}{2\theta}(u - v)^2 + \lambda \sum_{m \in I} |v - f^m| \quad (5.2)$$

must lie between  $u$  and  $\bar{f}$ . Consequently, the search for an optimal  $v$  (as described above) can be restricted by

$$\begin{aligned} l \in \{0, \dots, |I|/2\} & \quad \text{if} \quad u \leq \bar{f} \\ & \quad \text{and} \\ l \in \{|I|/2, \dots, |I|\} & \quad \text{if} \quad u \geq \bar{f}, \end{aligned}$$

which at least halves the computational effort.

### Recursive Computation

Using the fact that the distance fields  $f^m$ ,  $m \in I$ , are sorted in ascending order, a further speed up of the generalized thresholding step can be achieved by computing the sums  $\sum_{m \in I} |f^l - f^m|$  in (5.1) recursively:

**Theorem 5.1** (recurrence formula)

Let  $\{f^m : m \in I \subset \mathbb{N}\}$  be a set of real numbers with  $f^m \leq f^{m+1}$ . Then,

$$\sum_{m \in I} |f^{l+1} - f^m| = \sum_{m \in I} |f^l - f^m| + (2 \cdot l - |I|) |f^{l+1} - f^l|,$$

for all  $l = 1, \dots, |I| - 1$ .

**Proof:** Let  $l \in \{1, \dots, |I| - 1\}$ . Then, the sum  $\sum_{m \in I} |f^{l+1} - f^m|$  can be split by

$$\begin{aligned} \sum_{m \in I} |f^{l+1} - f^m| &= \sum_{m \leq l} \underbrace{|f^{l+1} - f^m|}_{\geq 0} + \sum_{m \geq l+1} \underbrace{|f^{l+1} - f^m|}_{\leq 0} \\ &= \sum_{m \leq l} (f^{l+1} - f^m) + \sum_{m \geq l+1} (f^m - f^{l+1}). \end{aligned}$$

Inserting  $0 = -f^l + f^l$  yields

$$\begin{aligned} \sum_{m \in I} |f^{l+1} - f^m| &= \sum_{m \leq l} (f^{l+1} - f^l + f^l - f^m) + \sum_{m \geq l+1} (f^m - f^l + f^l - f^{l+1}) \\ &= \sum_{m \leq l} \underbrace{(f^{l+1} - f^l)}_{\geq 0} + \sum_{m \leq l} \underbrace{(f^l - f^m)}_{\geq 0} + \sum_{m \geq l+1} \underbrace{(f^m - f^l)}_{\geq 0} + \sum_{m \geq l+1} \underbrace{(f^l - f^{l+1})}_{\leq 0} \end{aligned}$$

and by using the ascending order of the  $f^m$  we obtain

$$\begin{aligned} \sum_{m \in I} |f^{l+1} - f^m| &= \sum_{m \leq l} |f^{l+1} - f^l| + \sum_{m \leq l} |f^l - f^m| + \sum_{m \geq l+1} |f^l - f^m| - \sum_{m \geq l+1} |f^{l+1} - f^l| \\ &= \sum_{m \in I} |f^l - f^m| + l \cdot |f^{l+1} - f^l| - (|I| - l) \cdot |f^{l+1} - f^l| \\ &= \sum_{m \in I} |f^l - f^m| + (2 \cdot l - |I|) \cdot |f^{l+1} - f^l|. \end{aligned}$$

□

Consequently, only the first sum  $\sum_{m \in I} |f^1 - f^m|$  must be determined explicitly. All sums can be computed and stored once at the beginning of the reconstruction algorithm.

## 5.2 Experimental Validation

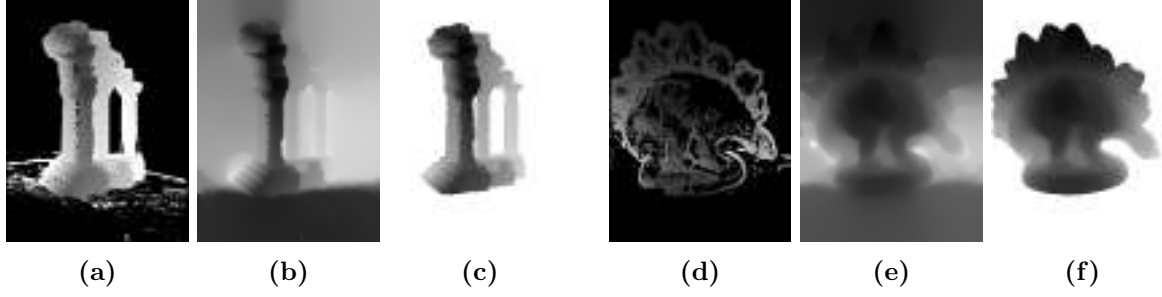
We tested the surface reconstruction algorithm on several multi-view data sets and obtained accurate 3D models for different kinds of objects. In the following, we present some of the results and analyze the influence of the used depth maps as well as the choice of parameters. All computational times were determined on a desktop PC equipped with a 3.3 GHz Intel Core i5-3550 processor and 8 GB RAM.

Name	Images	Image Size
<i>temple</i>	47	640×480
<i>dino</i>	48	640×480
<i>beethoven</i>	32	1024×768
<i>bunny</i>	35	1024×768

**Table 5.1:** Characteristics of the data sets used for our evaluation.



**Figure 5.3:** Images from the data sets in Table 5.1.



**Figure 5.4:** Comparison of depth map estimation methods. (a) and (d) Depth maps for the *temple* and *dino* created by correlation-based stereo matching. (b) and (e) Corresponding results obtained by multigrid stereo matching. (c) and (f) Restriction to the segmented foreground regions.

### 5.2.1 Data Sets

The different data sets used for our evaluation are briefly characterized in Table 5.1. A selection of corresponding images is displayed in Fig. 5.3.

The first two data sets, *temple* and *dino*, are the Middlebury “TempleRing” and “DinoRing” data sets discussed before. The temple object is well textured and exhibits small-scale details and sharp corners. In contrast, the dino figurine has a weakly textured smooth surface.

The data sets *beethoven* and *bunny* are provided by Klev et al. [32]. The Beethoven bust contains several parts that lack texture, while the bunny object presents mainly homogenous texture.

### 5.2.2 Depth Map Estimation

We used two different methods for depth map estimation, correlation-based stereo matching (corr) and multigrid stereo matching (mg).

The first method identifies corresponding points between two images: A similarity measure is computed and for each point in the first image, the point in the second image with the highest similarity is determined. We refer to [53, Chapter 3.10] for an introduction to stereo matching and similarity measures. The point correspondences can be used to extract 3D information and create a depth map. This stereo matching approach is very fast and works well in well textured areas. Weak texture and occlusions, however, yield erroneous matches that must be removed. Consequently, the resulting depth map is not dense but contains points without depth information.

Multigrid stereo matching [19] uses multigrid methods to solve a variational problem and obtain an optimal depth map. The variational model incorporates information about image intensities, image gradients and the smoothness of the depth map. Here, the 3D information for weakly textured regions is interpolated by using the surrounding structures. Hence, a value is assigned to each point, i.e., the resulting depth map is dense. Compared to the correlation-based approach the resolution is higher and therefore smaller objects can be reconstructed. However, the multigrid stereo matching is more complex and therefore also slower.

In Fig. 5.4 we present examples for depth maps obtained by correlation-based and multigrid stereo matching for images of the *temple* and *dino* data sets. Since the *temple* object is well textured, the correlation-based approach produces satisfying results; see Fig. 5.4a. Note that no distance values are assigned to the homogeneous background region. In contrast, the multigrid

approach interpolates values in weakly textured regions. Therefore, even the points representing the background and especially the areas between the pillars of the temple are filled up; see Fig. 5.4b. This is disadvantageous for the construction of distance fields (as described in Chapter 4.2) since these points are considered as surface points and would be reconstructed. To retain the free spaces between the pillars, we segmented the temple object as foreground in the input images and restricted the depth maps to this region; see Fig. 5.4c. As a side benefit, now fewer points must be considered for the distance field construction, which yields a significant speed-up. Since most of the images can easily be segmented into foreground and background regions, we proceeded similarly for the other data sets presented in Table 5.1 and thereby reduced the computation time for the conversion of depth maps into distance fields.

A correlation-based depth map for the *dino* is displayed in Fig. 5.4d. Owing to weak texture, this method fails for large parts of the object. The corresponding depth map created by multigrid stereo matching and its reduction to the image foreground are shown in Figs. 5.4e and 5.4f. Here, lowly textured parts are interpolated and dense depth information is obtained for the entire surface of the object. The higher quality of the multigrid depth map estimation, however, entails also higher runtimes. For both the *temple* and the *dino* data sets, each correlation-based depth map could be obtained within 0.03 seconds while the multigrid stereo matching took around 0.5 seconds. Consequently, for the choice of the right depth map estimation method, quality and computation time must be taken into account.

### 5.2.3 Results

In the following, we analyze the convergence behavior of the algorithm introduced in Chapter 4.3. Moreover, we present some results for the data sets in Table 5.1 and show how accuracy and computation time of the entire surface reconstruction process depend on the quality and number of depth maps as well as the voxel resolution. For the denoising step (Algorithm 1) we used  $\tau = 0.125$  and  $\epsilon = 0.00005 \cdot \# \{\text{voxels}\}$  for all data sets.

#### Convergence

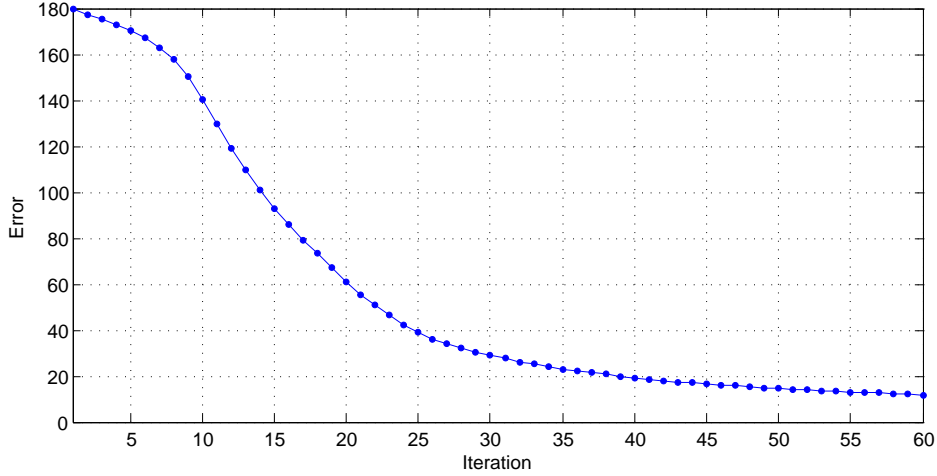
In general, the alternating optimization scheme converges fast for voxels with defined distance values. In regions with missing data, however, pure denoising (Algorithm 1) is performed, which yields slow convergence in these parts. In Fig. 5.5, the error

$$\max \left( \|u^{k-1} - u^k\|, \|v^{k-1} - v^k\| \right)$$

is plotted for the *temple* data set. It can be seen that the algorithm converges quickly at the beginning, but then slows down. Nevertheless, after a certain number of iterations the values for points that lie on the surface do not change any more and smoothing is only performed in points at the boundary of the voxel grid. Fig. 5.6 shows some intermediate results for the reconstruction of the *temple*. After around 18 iterations the resulting surface is virtually constant. Therefore, the reconstruction can be stopped after few iterations.

#### Influence of the Weight Parameters $\lambda$ and $\theta$

The parameter  $\lambda$  controls the occurrence of isolated points in the reconstruction. The lower  $\lambda$ , the more small detached features are removed. This behavior is illustrated in Fig. 5.7. The choice of this weight parameter highly depends on the quality of the underlying depth maps. Usually, for dense depth maps  $\lambda$  mainly influences the points at the boundary of the surface.



**Figure 5.5:** Convergence of the alternating optimization scheme for the *temple* data set. The error is calculated as  $\max(\|u^{k-1} - u^k\|, \|v^{k-1} - v^k\|)$  for each iteration  $k$ . The corresponding depth maps were obtained by multigrid stereo matching.



**Figure 5.6:** Intermediate results of the reconstruction algorithm. From left to right: Results after 5, 10, 18, 25, and 40 iterations.

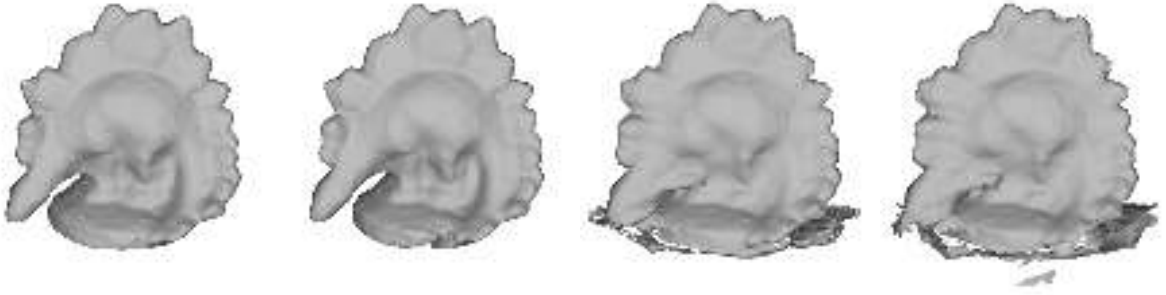
Here, lowering the value of  $\lambda$  results in a reduction of marginal noise, as can be seen in Fig. 5.7. For sparse depth maps, however, isolated matches in the depth maps might correspond to isolated 3D points. Therefore,  $\lambda$  must not be chosen too small, in order to retain these surface points.

Recall the optimization problem

$$\min_{(u,v) \in X \times X} J(u) + \frac{1}{2\theta} \|u - v\|_X^2 + \lambda G_F(v), \quad (\text{P}_\theta)$$

which is solved by alternately minimizing with respect to  $u$  and  $v$  separately, see Chapter 4.3.3. In each iteration of the reconstruction algorithm  $v$  presents an approximation of the distance fields  $f^m$  and  $u$  is obtained by a variational smoothing of  $v$ . The final surface is extracted from the zero set of the distance field  $u$ . The parameter  $\theta$  controls how close  $u$  is to  $v$ . Consequently, the lower  $\theta$  the more similar  $u$  is to  $v$ . Conversely, the higher  $\theta$  the more smoothing is performed during the denoising step, which results in a smoother surface. Fig. 5.8 shows some results for





**Figure 5.7:** The influence of the weight parameter  $\lambda$ . Reconstruction results for the *dino* data set. The values of  $\lambda$  from left to right: 0.08, 0.1, 0.5, 1.0. The corresponding depth maps were obtained by multigrid stereo matching.



**Figure 5.8:** The influence of the weight parameter  $\theta$ . Reconstruction results for the *temple* data set. The values of  $\theta$  from left to right: 0.01, 0.02, 0.03, 0.04. The corresponding depth maps were obtained by multigrid stereo matching.

the *temple* data set for different values of  $\theta$ . In the following, we set the value of  $\theta$  to 0.02 for all data sets.

### Reconstruction Results

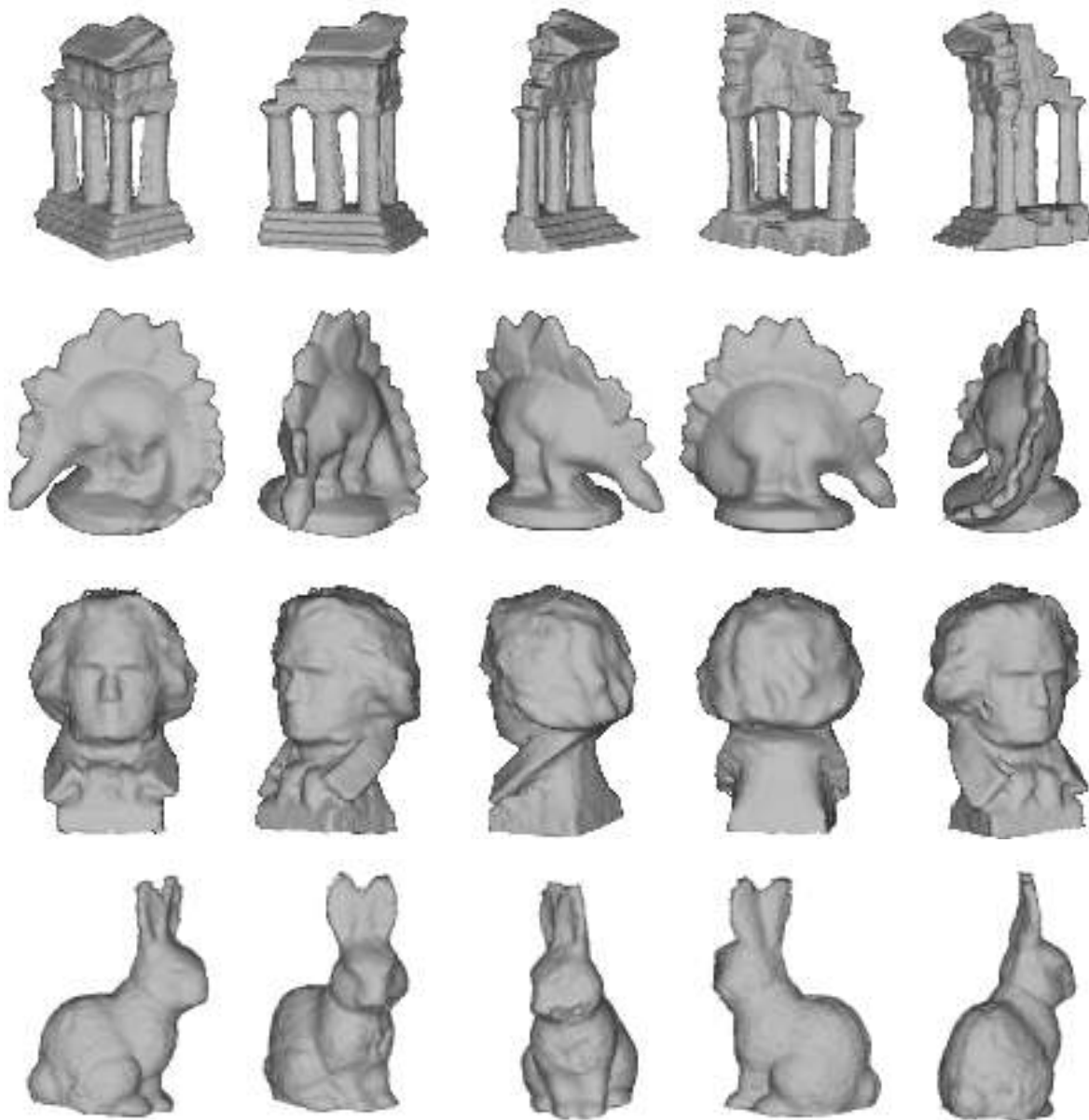
In Fig. 5.9, we present some images of our reconstruction results for the data sets in Table 5.1. The corresponding characteristics as well as the computation times for the individual steps of the reconstruction process are listed in Table 5.2.

The Middlebury data sets *temple* and *dino* are perfectly suited for depth map based surface reconstructions: The illumination is nearly constant and the motion between two succeeding images is sufficiently small, which facilitates the depth map estimation.

As discussed in Section 6.3, the well-textured surface of the temple object yields high-quality depth maps. Consequently, the corresponding reconstructions are very accurate. Using multigrid stereo matching, we obtained a detailed 3D-model that exhibits small details and sharp features; see Fig. 5.9. We also tested the reconstruction algorithm on the same data set with correlation-based stereo matching. Even though the correlation-based depth map estimation is faster than the multigrid stereo matching, the overall computation time for the correlation-based reconstruction is higher, since more smoothing has to be performed at points with missing data.

Name	Depth Maps	Voxels	Depth Map Estimation	Distance Field Generation	Reconstruction (Iterations)	Total
<i>temple</i>	46 (mg)	$205 \times 321 \times 151$	23s	2m15s	2m51s (18)	5m29s
<i>dino</i>	47 (mg)	$183 \times 219 \times 185$	23s	1m47s	1m54s (18)	4m5s
<i>beethoven</i>	23 (mg)	$252 \times 302 \times 377$	46s	9m12s	11m59s (100)	21m57s
<i>bunny</i>	16 (mg)	$216 \times 287 \times 323$	22s	1m37s	6m4s (100)	8m3s

**Table 5.2:** Characteristics of the depth maps used for the reconstruction of the data sets in Table 5.1 and runtimes of the individual steps of the reconstruction process.

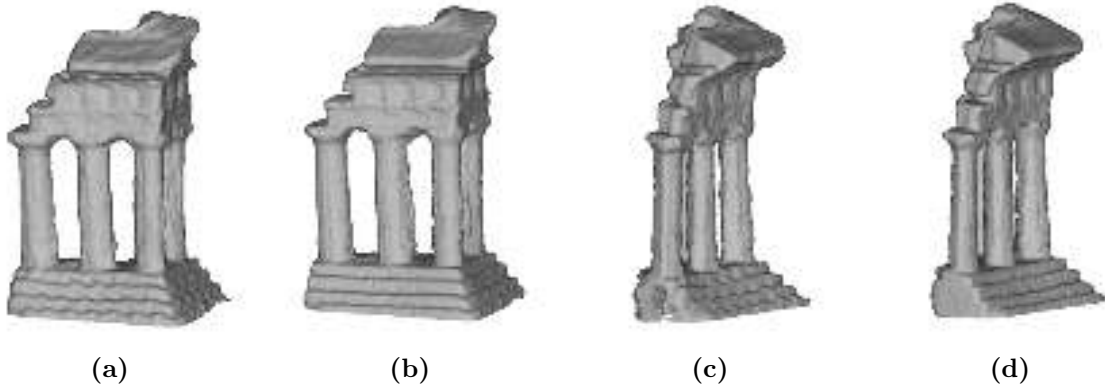


**Figure 5.9:** Our reconstruction results for the data sets in Table 5.1. All corresponding depth maps were obtained by multigrid stereo matching.

Name	Depth Maps	Voxels	Depth Map Estimation	Distance Field Generation	Reconstruction (Iterations)	Total
<i>temple</i>	46 (mg)	$205 \times 321 \times 151$	23s	2m15s	2m51s (18)	5m29s
<i>temple</i>	46 (corr)	$205 \times 321 \times 151$	1s	2m16s	4m47s (20)	7m4s
<i>dino</i>	47 (mg)	$183 \times 219 \times 185$	23s	1m47s	1m54s (18)	4m5s
<i>dino</i>	47 (corr)	$183 \times 219 \times 185$	1s	1m47s	3m15s (30)	5m2s

**Table 5.3:** Comparison of the runtimes of reconstructions with depth maps generated by multigrid and correlation-based stereo matching for the Middlebury data sets.

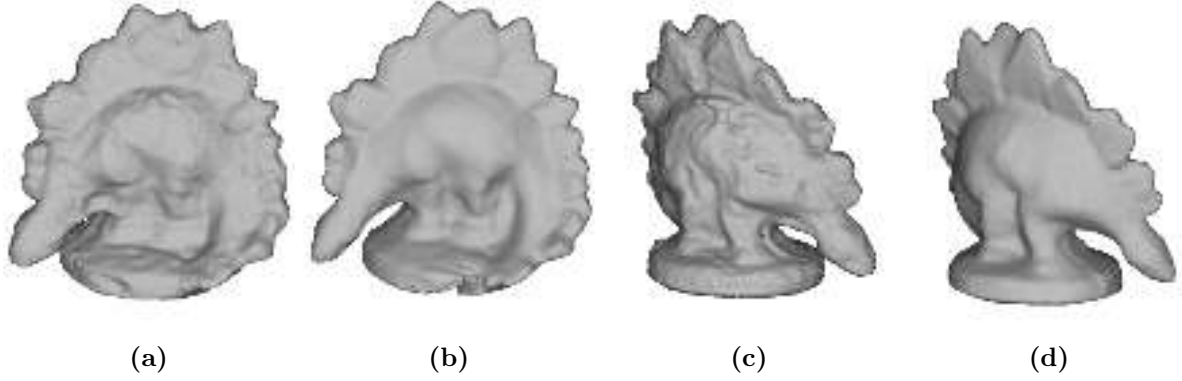
A comparison of the runtimes for the individual reconstruction steps is presented in Table 5.3. Compared to the results obtained by multigrid stereo matching the surfaces obtained by using correlation-based depth maps is less detailed and the edges are less sharp; see Fig. 5.10. In both cases  $\lambda$  was set to 0.16.



**Figure 5.10:** Comparison of the reconstructions for the *temple* data set received by using different depth maps. (a) and (c) Results obtained with correlation-based depth maps. (b) and (d) Results obtained with depth maps constructed by multigrid stereo matching.

The influence of the quality of the depth maps is more obvious for the *dino* data set. Fig. 5.9 shows our result obtained by using multigrid stereo matching for the construction of depth maps. As discussed in Section 6.3, correlation-based stereo matching for the images of the *dino* data set results in sparse depth maps. Since many values are missing for the homogenous regions of the surface, smoothing has to be performed in these areas, which slows down the reconstruction process, see Table 5.3. The corresponding result exhibits an irregular surface and can not compete with the 3D-model received by using dense depth maps (mg), as can be seen in Fig. 5.11. We used  $\lambda = 0.1$  for the correlation-based reconstruction and  $\lambda = 0.08$  for the one based on multigrid stereo matching.

The motions between two succeeding images of the *beethoven* and the *bunny* data sets are much bigger than for the Middlebury data sets. Owing to occlusions and lightning changes, many erroneous matchings occur during the generation of depth maps. For some image pairs, the depth estimation even fails totally. We removed the corresponding depth maps and therefore used only parts of the data. For both data sets multigrid stereo matching was used to compensate occlusions by interpolation of distance values.



**Figure 5.11:** Comparison of the reconstructions for the *dino* data set received by using different depth maps. (a) and (c) Results obtained with correlation-based depth maps. (b) and (d) Results obtained with depth maps constructed by multigrid stereo matching.

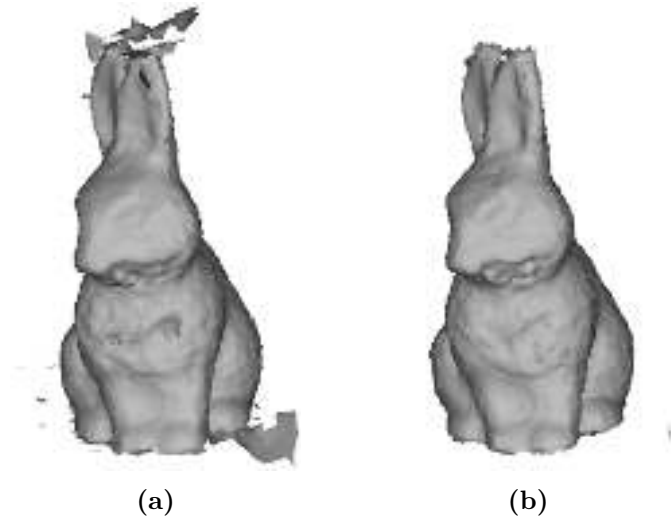
We chose 23 depth maps for the *beethoven* data set. Both, the size ( $252 \times 30 \times 377$ ) of the underlying voxel grid and the poor quality of the depth maps entail high computation times for the reconstruction process; see Table 5.2. Nevertheless, even the low-textured parts like the cheeks of the Beethoven bust can be reconstructed; see Fig. 5.9. The value of  $\lambda$  was set to 0.3.

For the surface reconstruction for the *bunny* data set, we used 23 depth maps in a first try. There are two critical parts of the bunny object, the chest and the back. The back is only visible in a few images, while the chest appears under different lightning conditions. Therefore, erroneous matchings are present in several depth maps, which yield dents in the corresponding reconstructions. To overcome this problem, we removed the depth maps obtained from the front view. Using the resulting 16 depth maps, the surface at the critical points at the chest was then obtained by almost pure smoothing, since only few data remained in these parts. A comparison of the results obtained with 23 and 16 depth maps is shown in Fig. 5.12. More views of the latter are presented in Fig. 5.9 and the corresponding computation times are listed in Table 5.2. We used  $\lambda = 0.3$  for the reconstruction algorithm.

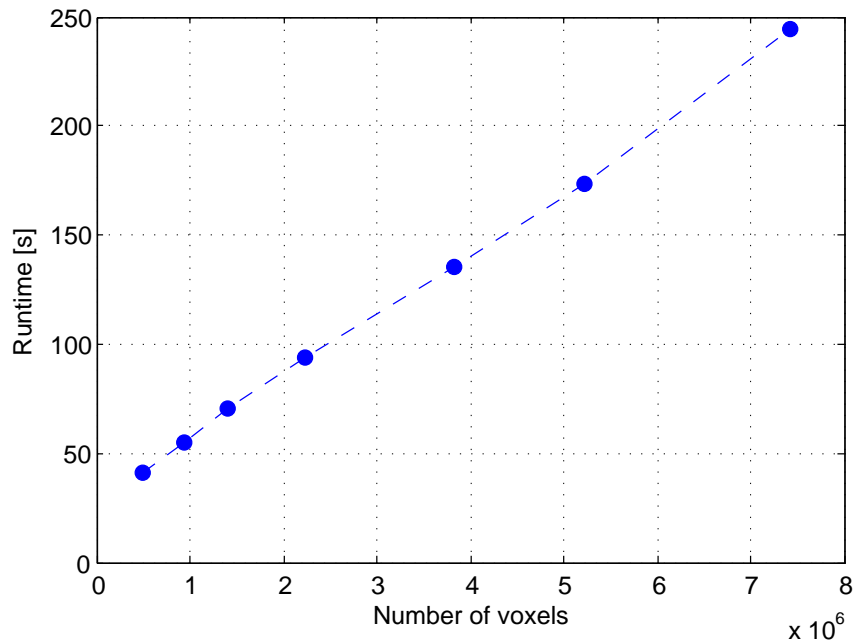
### Voxel Resolution

The entire reconstruction algorithm is linear in the size of the underlying voxel grid, as can be seen in Fig. 5.13. The results for different resolutions for the *dino* data set are presented in Fig. 5.14. Note that reducing the resolution to  $92 \times 110 \times 93$  voxels still yields an exact representation of the shape of the object. The corresponding computation time reduces to 54 seconds. Hence, if only the shape of an object is required, a reconstruction within less than one minute is possible.

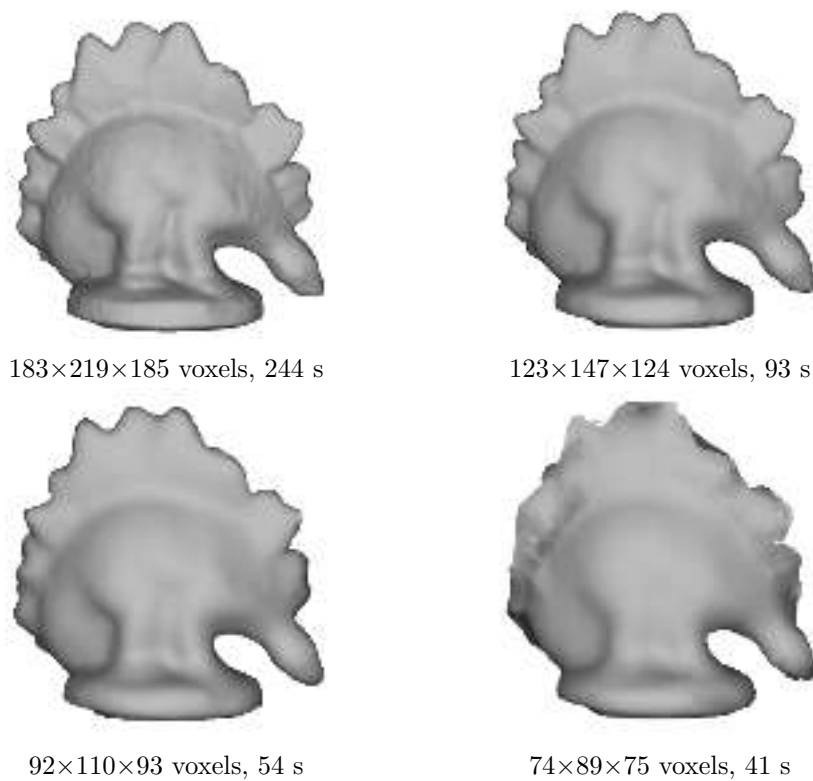
Conclusively, we want to remark that wide parts of the presented surface reconstruction algorithm are suitable for a parallel implementation. Thus, by parallelizing several computations, a considerable speed up of the reconstruction process can be expected. In the following chapter we will address further possibilities of improving the performance of the algorithm.



**Figure 5.12:** Comparison of the reconstructions for the *bunny* data set received by using different numbers of depth maps. (a) Result obtained with 23 depth maps (mg). (b) Result obtained after the reduction to 16 depth maps (mg).



**Figure 5.13:** The runtime of the reconstruction algorithm for different resolutions of the underlying voxel grid for the *dino* data set (mg).



**Figure 5.14:** Reconstruction results and runtimes for the *dino* data set with different resolutions of the underlying voxel grid. The corresponding depth maps were obtained by multigrid stereo matching.

We have presented a depth-map-based algorithm for multi-view stereo reconstruction and proved its convergence to an optimal solution. The method was tested on several different data sets and convincing results could be obtained within a few minutes for highly and lowly textured objects as well as large-scale scenes. The algorithm outperforms most other reconstruction techniques in terms of computation time. However, the accuracy and performance of the entire reconstruction process highly depends on the used depth map estimation method.

To complete this work, we will now present possible modifications of the proposed variational approach that can be used to reduce the computation time of the reconstruction method in order to make it more suitable for time-critical applications. Furthermore, we will discuss applications of the TV- $L^1$  energy model to other image processing tasks, including depth map estimation.

## 6.1 GPU-friendly Approximation of the Energy Functional

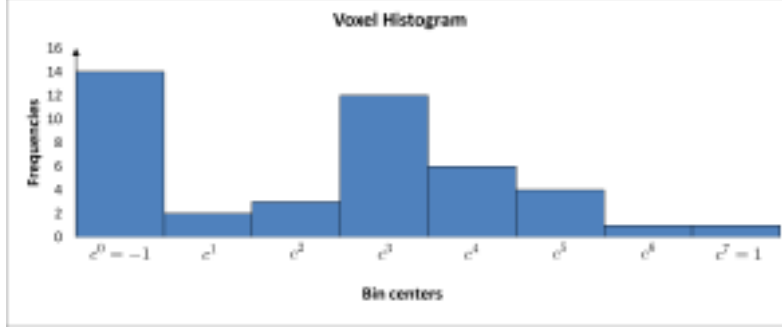
Numerous image processing and computer vision algorithms can be parallelized and therefore accelerated by a GPU. The reconstruction algorithm (Algorithm 3), however, is not well suited for a GPU-based implementation: Recall the generalized thresholding step in Algorithm 2. The number of iterations needed to compute the optimal value  $v_{i,j,k}$  varies for each voxel, which is disadvantageous for an efficient parallelization. To overcome this problem, Zach [59] introduced a GPU-friendly approximation of the energy functional

$$\min_{u \in BV(\Omega)} J(u) + \lambda \int_{\Omega} \sum_{m \in I(x)} |u(x) - f^m(x)| dx, \quad (Z)$$

as described next. The interval  $[-1, 1]$  is partitioned into a fixed number of bins with evenly spaced bin centers  $c^n$ . Then, for each voxel  $x$  the values of  $f^m$ ,  $m \in I$ , are assigned to the corresponding bins and a histogram is obtained; see Fig. 6.1.

Replacing each  $f^m$  by its closest bin center  $c^n$  yields the histogram-based energy functional

$$E^{\text{Hist}} = J(u) + \lambda \int_{\Omega} \sum_n r^n(x) |u(x) - c^n| dx, \quad (6.1)$$



**Figure 6.1:** Histogram of distance values for one voxel. The interval  $[-1, 1]$  is partitioned into several bins and each distance value is assigned to its corresponding bin.

where  $r^n$  denotes the number of distance fields that lie in bin  $n$ . To find an (approximate) minimizer of  $E^{\text{Hist}}$ , a modification of the alternating optimization scheme in Chapter 4.3.3 can be used: Since the regularization term did not change, the denoising problem remains the same, but for  $u_{i,j,k} \in (c^n, c^{n+1})$  the generalized thresholding step is replaced by

$$v_{i,j,k} = \max \left\{ c^n - \epsilon, \min \left\{ c^{n+1} + \epsilon, v_{i,j,k}^* \right\} \right\},$$

with

$$v_{i,j,k}^* = u_{i,j,k} + \lambda \theta \left( \sum_{l \geq n} r_{i,j,k}^l - \sum_{l \leq n} r_{i,j,k}^l \right)$$

and a small  $\epsilon > 0$ . Hence, the amount of computations is the same for each voxel and the resulting modified algorithm can be performed on a GPU. Comparing the results of the accelerated method (“Zach2”) to the original one, Zach [59] accomplished a reduction of computation time by a factor of 12 on the Middlebury “Ring” data sets; see Figs. 1.8a and 1.9a. Additionally, the high quality of the corresponding reconstructions is retained or even increased (see Figs. 1.8b and c and Figs. 1.9b and c), which underlines the power of this modification.

## 6.2 Variational Denoising Revisited

Recall the ROF-denoising problem introduced in Chapter 3.2:

$$\min_{u \in BV(\Omega)} J(u) + \frac{1}{2\theta} \|u - f\|_2^2. \quad (\text{ROF})$$

This model has been used for noise reduction in the context of various image processing applications and therefore has been thoroughly discussed during the last decade. Several variants have been proposed and many different optimization schemes have been developed.

Solving (ROF) is the first step in each iteration of the reconstruction process (Algorithm 3). For this purpose, Zach et al. used a denoising method introduced by Chambolle in 2004; see Chapter 3.4. Recently, however, more efficient optimization schemes have been developed. We will explain the ideas of two such methods that (possibly in combination with the histogram-based data fidelity described above) could replace Chambolle’s fixed point algorithm to further accelerate the surface reconstruction.



### The Split Bregman Method

Goldstein et al. [24] introduced a method that is based on the *Bregman iteration technique* [8] for convex optimization: Let  $E$  and  $H$  be convex functions on  $\mathbb{R}^n$ . Furthermore, let  $H$  be differentiable and  $\min_u H(u) = 0$ . Then,

$$\min_u E(u) + \lambda H(u)$$

can be solved by iteratively solving the sub-problems

$$u^{k+1} = \min_u E(u) - \langle p^k, u - u^k \rangle + \lambda H(u),$$

with  $\langle \cdot \rangle$  the euclidean inner product and

$$p^{k+1} = p^k - \nabla H(u^{k+1}).$$

Goldstein et al. adapted this method to the denoising model (ROF) and developed a very efficient solver. This so-called *Split Bregman* algorithm uses the Gauss-Seidel method to solve the sub-problems and can easily be parallelized. It outperforms Chambolle's algorithm and most other solvers for total variation denoising.

### Proximity Algorithm

In 2011, Micchelli et al. [41] proposed a denoising algorithm that is based on the *proximity operator*. Let  $F$  be a real-valued function. Then the proximity operator of  $F$  is defined by

$$\text{prox}_{\theta F}(v) := \arg \min_u F(u) + \frac{1}{2\theta} \|u - v\|_2^2, \quad \theta > 0.$$

Consequently, the solution of (ROF) is the proximity operator of the total variation functional  $J$  evaluated at the noisy data  $v$ . Micchelli et al. derived a general fixed-point method to determine  $\text{prox}_{\varphi \circ B}$ , for a convex function  $\varphi$  and a linear transformation  $B$ . Since the discrete total variation (see Chapter 3.3.2) is the composition of the convex euclidean norm and the linear gradient operator, this algorithm can be applied to total variation denoising. Experiments show that the corresponding solver needs fewer iterations in comparison to the Split Bregman technique.

It can be proved that all three methods, Chambolle's algorithm, the Split Bregman algorithm, and the proximity algorithm, compute the projection of  $v$  onto the convex set  $\theta K$  (see Chapter 3.4.3) in different ways.

## 6.3 Additional Uses of the TV- $L^1$ -Energy Model

Zach et al. [61] modeled the surface reconstruction process as finding an optimal distance field that solves

$$\min_{u \in BV(\Omega)} J(u) + \lambda \int_{\Omega} \sum_{m \in I(x)} |u(x) - f^m(x)| dx, \quad (\text{Z})$$

where the distance fields  $f^m$  are constructed from input depth maps  $r_m$  as described in Chapter 4.2. Note that the proposed optimization algorithm in Chapter 4.3.3 does not depend on the actual way of converting depth maps to distance fields. Therefore, it can be used for general distance maps. Moreover, with only slight modifications, the optimization scheme can also be used for other applications like optical flow and depth map estimation, as described next.

## Optical Flow

Let  $I_1$  and  $I_2$  be two-dimensional images. The optical flow describes the motion of pixels between  $I_1$  and  $I_2$  and can be described by a displacement field  $u(x) = (u_1(x), u_2(x))$  defined on the image domain. An optimal flow  $u$  can be found by solving the TV- $L^1$  problem

$$\min_{u=(u_1, u_2)} J(u_1) + J(u_2) + \lambda \int_{\Omega} |I_2(x + u(x)) - I_1(x)| dx,$$

as proposed by Zach et al. [60]. Let  $u_0$  be a given displacement field. Then, near  $u_0$ ,  $I_2(x + u(x))$  can be replaced by its first order Taylor approximation. Define

$$\rho(u, u_0, x) := I_2(x + u_0) + \langle \nabla I_2, u - u_0 \rangle - I_1(x).$$

Then, we obtain the optimization problem

$$\min_{u=(u_1, u_2)} J(u_1) + J(u_2) + \lambda \int_{\Omega} |\rho(u, u_0, x)| dx$$

that can be solved by an alternating optimization scheme very similar to the one in Chapter 4.3.3: In a first step, variational denoising is performed for both  $u_1$  and  $u_2$ . The second step is a simple thresholding that depends on the value of  $\nabla I_2$ . The algorithm can be implemented efficiently on GPUs and real-time performance is possible. Moreover, this approach can be extended to general  $n$ -dimensional images.

## Depth Map Estimation

In 2010, Stühmer et al. [55] adopted this optical flow approach to depth map estimation from multiple images. Let  $I_i : \Omega_i \rightarrow \mathbb{R}$ ,  $\Omega_i \subset \mathbb{R}^2$ ,  $i = 0, \dots, K$ , be calibrated images. An optimal depth map should be consistent with all these input images and preserve sharp features of an object. Assuming Lambertian surfaces, the depth map estimation problem can be formulated as

$$\min_{h \in BV(\Omega)} J(h) + \lambda \int_{\Omega_0} \sum_{i \in \mathcal{I}(x)} |I_i(x, h) - I_0(x)| dx$$

where  $I_i(x, h)$  denotes the image intensity at the projection of a point  $x$  into image  $I_i$  using depth map  $h$ .  $\mathcal{I}(x)$  contains the indices of all these images for which the projection lies within the image domain. Again,  $I_i(x, h)$  can be linearized by Taylor expansion and we define

$$\rho_i(x, h) := I_i(x, h_0) + (h - h_0) \partial_h I_i(x, h) \Big|_{h_0} - I_0(x)$$

and obtain the optimization problem

$$\min_{h \in BV(\Omega)} J(h) + \lambda \sum_{i \in \mathcal{I}(x)} |\rho_i(x, h)| dx,$$

which can be solved by alternating total variation denoising and generalized thresholding in direct analogy to Chapter 4.3.3. Here, the thresholding step depends on the critical points where  $\rho_i(x, h)$  equals zero.

Stühmer et al. embedded this method in a camera tracking approach and succeeded in obtaining dense depth maps from a handheld camera in real-time. Using more than two images for depth

map estimation reduces the amount of noise contained in the resulting depth map.

Combining this fast and robust construction of depth maps with the accelerated surface reconstruction algorithm presented in Section 6.1 might be another step towards high quality multi-view stereo reconstruction within only a few seconds.



---

## List of Figures

---

1.1	Examples of 3D model representations . . . . .	2
1.2	The reconstruction of Michelangelo’s David realized by a research group from Stanford University . . . . .	2
1.3	The concept of multi-view stereo reconstruction . . . . .	4
1.4	The Middlebury data sets . . . . .	5
1.5	The visual hull . . . . .	6
1.6	Examples of depth maps for two objects . . . . .	8
1.7	Surface reconstruction results for some of the state-of-the-art methods on the Middlebury data sets . . . . .	10
1.8	Quantitative comparison of several multi-view stereo reconstruction methods for the Middlebury “TempleRing” data set (47 views) . . . . .	11
1.9	Quantitative comparison of several multi-view stereo reconstruction methods for the Middlebury “DinoRing” data set (48 views) . . . . .	12
3.1	Variational image denoising . . . . .	23
3.2	An example of variational image denoising . . . . .	32
3.3	Variational image deblurring . . . . .	33
3.4	Variational inpainting . . . . .	34
4.1	Generation of distance fields from depth maps. . . . .	37
5.1	An example of run-length encoding . . . . .	48
5.2	An example of an octree . . . . .	49
5.3	Images from the data sets in Table 5.1. . . . .	51
5.4	Comparison of depth map estimation methods . . . . .	52
5.5	Convergence of the alternating optimization scheme . . . . .	54
5.6	Intermediate results of the reconstruction algorithm . . . . .	54
5.7	The influence of the weight parameter $\lambda$ . . . . .	55
5.8	The influence of the weight parameter $\theta$ . . . . .	55
5.9	Our reconstruction results for the data sets in Table 5.1 . . . . .	56
5.10	Comparison of the reconstructions for the <i>temple</i> data set . . . . .	57
5.11	Comparison of the reconstructions for the <i>dino</i> data set . . . . .	58
5.12	Comparison of the reconstructions for the <i>bunny</i> data set . . . . .	59

5.13	The runtime of the reconstruction algorithm for different resolutions of the underlying voxel grid . . . . .	59
5.14	Reconstruction results and runtimes for the <i>dino</i> data set with different resolutions of the underlying voxel grid . . . . .	60
6.1	Voxel histogram . . . . .	62

---

## List of Tables

---

5.1	Characteristics of the data sets used for our evaluation. . . . .	51
5.2	Characteristics of the used depth maps and runtimes of the reconstructions . . .	56
5.3	Comparison of reconstructions with depth maps generated by multigrid and correlation-based stereo matching . . . . .	57





---

## Bibliography

---

- [1] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Clarendon Press, Oxford, 2000.
- [2] A. Auclair, N. Vincent, and L. D. Cohen. Using point correspondences without projective deformation for multi-view stereo reconstruction. In *IEEE 15th International Conference on Image Processing*, pages 193–196, 2008.
- [3] J.-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *Journal of Mathematical Imaging and Vision*, 22(1):71–88, 2005.
- [4] J.F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition-modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.
- [5] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer, New York, 2000.
- [6] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [7] K. Bredies and D. Lorenz. *Mathematische Bildverarbeitung*. Vieweg+Teubner, Wiesbaden, 2011.
- [8] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.
- [9] N. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision–ECCV 2008*, pages 766–779. Springer, 2008.
- [10] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.

- [11] 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 (Radon Series on Computational and Applied Mathematics)*, 9:263–340, 2010.
- [12] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.
- [13] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen. An augmented lagrangian method for total variation video restoration. *IEEE Transactions on Image Processing*, 20(11):3097–3111, 2011.
- [14] T. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 2005.
- [15] J. Y. Chang, H. Park, I. K. Park, K. M. Lee, and S. U. Lee. GPU-friendly multi-view stereo reconstruction using surfel representation and graph cuts. *Computer Vision and Image Understanding*, 115(5):620–634, 2011.
- [16] R. T. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.
- [17] A. Delaunoy, E. Prados, P. G. I. Piracés, J.-P. Pons, and P. Sturm. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *British machine vision conference*, 2008.
- [18] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, and Y. Wang. Noisy depth maps fusion for multi-view stereo via matrix completion. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):566–582, 2012.
- [19] B. Drost. Ein variationsbasiertes Verfahren zum Stereosehen. Diplomarbeit, Fakultät für Mathematik, Technische Universität München, 2008.
- [20] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, Rhode Island, second edition, 1998.
- [21] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.
- [22] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.
- [23] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
- [24] T. Goldstein and S. Osher. The split Bregman method for  $L^1$ -regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
- [25] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

- 
- [26] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, second edition, 2003.
  - [27] P. S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.
  - [28] C. Hernández Esteban and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.
  - [29] V. H. Hiep, R. Keriven, P. Labatut, and J. P. Pons. Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1437, 2009.
  - [30] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer, Berlin Heidelberg, 1993.
  - [31] A. Hornung and L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 503–510, 2006.
  - [32] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3D reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009.
  - [33] K. N Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.
  - [34] M.-J. Lai, B. Lucier, and J. Wang. The convergence of a central-difference discretization of Rudin-Osher-Fatemi model for image denoising. In *Scale Space and Variational Methods in Computer Vision*, pages 514–526. Springer, 2009.
  - [35] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.
  - [36] M. Levoy. The digital Michelangelo project. <https://graphics.stanford.edu/projects/mich/>, August 2009. Accessed May 27, 2013.
  - [37] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, et al. The digital Michelangelo project: 3D scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000.
  - [38] J. Li, E. Li, Y. Chen, L. Xu, and Y. Zhang. Bundled depth-map merging for multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2769–2776, 2010.
  - [39] Y. Liu, X. Cao, Q. Dai, and W. Xu. Continuous depth estimation for multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2121–2128, 2009.
  - [40] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. M. Frahm, R. Yang, D. Nistér, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
  - [41] C. A. Micchelli, L. Shen, and Y. Xu. Proximity algorithms for image models: denoising. *Inverse Problems*, 27(4):045009, 2011.

- [42] M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120, 2004.
- [43] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.
- [44] J. P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 822–827, 2005.
- [45] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
- [46] D. Scharstein. The Middlebury computer vision pages. <http://vision.middlebury.edu/mview/>, April 2012. Accessed May 27, 2013.
- [47] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.
- [48] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–528. IEEE, 2006.
- [49] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.
- [50] S. N. Sinha, P. Mordohai, and M. Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
- [51] S. Soatto, A. J. Yezzi, and H. Jin. Tales of shape and radiance in multiview stereo. In *9th IEEE International Conference on Computer Vision*, pages 974–981, 2003.
- [52] M. Sormann, C. Zach, J. Bauer, K. Karner, and H. Bishof. Watertight multi-view reconstruction based on volumetric graph-cuts. In *Image Analysis*, pages 393–402. Springer, 2007.
- [53] C. Steger, M. Ulrich, and C. Wiedemann. *Machine Vision Algorithms and Applications*. WILEY-VCH, Weinheim, 2008.
- [54] C. Strecha, R. Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2394–2401, 2006.
- [55] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition*, pages 11–20. Springer, 2010.
- [56] C. J. Taylor. Surface reconstruction from feature based stereo. In *9th IEEE International Conference on Computer Vision*, pages 184–190, 2003.
- [57] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, 2005.

- [58] J. Wang and B. J. Lucier. Error bounds for finite-difference methods for Rudin-Osher-Fatemi image smoothing. *SIAM Journal on Numerical Analysis*, 49(2):845–868, 2011.
- [59] C. Zach. Fast and high quality fusion of depth maps. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, volume 1, 2008.
- [60] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV- $L^1$  optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007.
- [61] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV- $L^1$  range image integration. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
- [62] A. Zaharescu, E. Boyer, and R. Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *Computer Vision–ACCV 2007*, pages 166–175. Springer, 2007.