TECHNISCHE UNIVERSITÄT MÜNCHEN

Department of Computer Science

Computer Vision Group

Convex Variational Methods for Single-View and Space-Time Multi-View Reconstruction

Martin Ralf Oswald

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Lehrstuhl für Computer Vision and Pattern Recognition

Convex Variational Methods for Single-View and Space-Time Multi-View Reconstruction

Martin Ralf Oswald

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

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Nassir Navab
Prüfer der Dissertation:	
1.	Univ.-Prof. Dr. Daniel Cremers
2.	Univ.-Prof. Dr. Marc Pollefeys ETH Zürich, Schweiz

Die Dissertation wurde am 28.10.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 07.02.2015 angenommen.

Abstract

This thesis investigates two special cases of 3D reconstruction: The reconstruction from only a single input image and the reconstruction over time from multiple-view image sequences. For both cases we propose several mathematical models which share the same basic idea: We compute a surface with minimal area that best fits the input data and suitable priors as the solution of a variational, convex optimization problem. Compared to state-of-the-art methods the proposed single-view reconstruction approaches require significantly less user input and yield competitive reconstruction results. For the space-time multi-view case, we show that the quality of a temporally coherent reconstruction improves substantially compared to time-independent approaches. A novel data fidelity term, the estimation and integration of surface normals and the integration of a novel, generalized form of connectivity constraints leads to reconstruction results that outperform the state of the art in both reconstruction accuracy and computation time.

keywords: single-view reconstruction, spatio-temporal multi-view reconstruction, minimal surfaces, convex optimization, shape priors, volume priors, connectivity constraints

Kurzfassung

Diese Arbeit untersucht zwei Spezialfälle der 3D Rekonstruktion: Die Rekonstruktion von einem einzigen Eingabebild, sowie die räumlich-zeitliche Rekonstruktion anhand von Bildsequenzen mehrerer Kameras. Für beide Szenarien schlagen wir mehrere mathematische Modelle vor, die der gleichen Idee zugrunde liegen: Wir berechnen eine Minimalfläche, welche die Eingabedaten und geeignete a-priori Annahmen am besten erfüllt, als Lösung eines variationellen, konvexen Optimierungsproblems. Im Vergleich zu anderen aktuellen single-view Verfahren benötigt die vorgeschlagene Methode deutlich weniger Benutzereingaben und erzielt dabei Ergebnisse, die sich mit dem Stand der Technik messen können. Für den Fall der räumlich-zeitlichen Rekonstruktion zeigen wir, dass sich die Qualität einer zeitlich-kohärenten Rekonstruktion im Vergleich zu einer zeitlich unabhängigen Rekonstruktion deutlich verbessert. Ein neuer Datenterm, die Schätzung und Integration von Flächennormalen, sowie die Integration einer neuen, generalisierten Form von Konnektivitätsbedingungen führen zu Rekonstruktionsergebnissen, die aktuelle vergleichbare Verfahren in Bezug auf Genauigkeit und Rechenzeit übertreffen.

Stichworte: Einzelbildrekonstruktion, räumlich-zeitliche Rekonstruktion, Minimalflächen, konvexe Optimierung, Gestaltannahmen, Volumenannahmen, Konnektivitätsbedingungen

Summary

Recovering three-dimensional geometry from a set of color images is a central problem in computer vision and has a wide range of applications. This thesis contributes approaches to two extreme cases of this task. The reconstruction of 3D geometry from a single input image only, referred to as *Single-View Reconstruction*, and the 3D reconstruction of a dynamic scene from a set of multiple, simultaneously filmed videos, referred to as *Spatio-Temporal Multi-View Reconstruction*.

For both cases we propose several mathematical models which share the same basic idea. We compute a surface with minimal area that best fits the input data and suitable priors as the solution of a variational, convex optimization problem. The frequent occurrence of minimal surfaces in nature and man-made objects as well as their elegant mathematical formulation with several desirable properties constitutes them as a reasonable and attractive prior assumption for this task. We further propose and study additional shape, volume, symmetry and connectivity priors to guide the reconstruction process in the two different scenarios which are discussed separately in the following.

Single-View Reconstruction is the most difficult case in image-based reconstruction, because correspondences between multiple input images cannot be utilized to recover the depth information that is lost due to the projective mapping. Instead of recovering a view-dependent depth representation of the scene, we aim to estimate full non-exact but plausible 3D models with the help of novel priors and a small amount of user input. We propose three different models for interactive single-view reconstruction which demonstrate the effectiveness of the minimal surface prior in conjunction with 1) a reflective planar symmetry prior to recover the back side of the object; and 2) either an explicit shape prior or a volume prior to inflate the scene into the third dimension. Due to a non-parametric surface representation, the solutions of these models can have arbitrary topology and are either globally optimal or within small bounds of the optimal solution. All models require significantly less user input and the reconstruction results compare well to state-of-the-art methods.

Spatio-Temporal Multi-View Reconstruction generalizes the problem of 3D reconstruction from multiple images of a static scene to the time domain. That is, the goal is a temporally coherent 3D reconstruction of a dynamic scene from multiple input videos, for instance, human motion over time. In a sense the problem is opposite to the single view case, because it deals with huge amounts of input data. With the reasonable assumption that dynamic scenes only change slowly over time we show that the quality of a temporally coherent reconstruction improves compared to a time-independent approach by leveraging information from consecutive time steps. Building on existing work on static multi-view 3D reconstruction, several extensions and improvements are suggested and evaluated, demonstrating that the proposed minimal surface approach outperforms state-of-the-art reconstruction methods in quality and speed.

Acknowlegdements

First and foremost, I want to thank my supervisor, Prof. Daniel Cremers, for giving me the opportunity to pursue my PhD under his supervision and for guiding me into research. He helped, inspired, guided and challenged me throughout the course of this thesis. Moreover, I learned several secondary skills from him, in particular, how to present research results convincingly and how to write things down concisely and understandable.

Further, I want to thank my examining committee: Prof. Nassir Navab, Prof. Daniel Cremers and Prof. Marc Pollefeys for their interest in this topic and reading my thesis.

The contents of this thesis have grown in collaboration with several people and this work would certainly not have been possible without my co-authors. Therefore, I want to thank especially Eno Töppe with whom I closely collaborated on the topic of single-view reconstruction and who contributed a lot to make the start of my journey into research interesting, pleasant and fruitful. Further, I thank my other co-authors Jan Stühmer, Claudia Nieuwenhuis, Tobias Gurdan, Kalin Kolev, Carsten Rother and Daniel Cremers for their contributions and collaboration.

I also want to thank my various office mates for helping with word and deeds and starting or joining discussions on both research-related and -unrelated topics. These include Eno Töppe, Thomas Windheuser, Matthias Vestner, Emanuele Rodolà, Youngwook Kee, Jan Stühmer, Caterina Vitadello, Kalin Kolev and Bastian Goldlücke.

In a similar way, all my other co-workers made may time in the group in one way or the other fruitful and enjoyable: Julia Diebold, Jakob Engel, Michael Karg, Christian Kerl, Maria Klodt, Thomas Möllenhoff, Mohamed Souiai, Frank Steinbrücker, Evgeny Strekalovskiy, and Jürgen Sturm.

I am grateful to Steffen Jaensch, Matthias Vestner and Georgiana for proof reading parts of this thesis and giving helpful and constructive suggestions for improvements. I also thank Frank Steinbrücker for proof reading most of my research papers and for many interesting and funny conversations about anything that crossed our minds.

Lastly, I want to thank my family and especially Georgiana for their help, love and support in everything I was doing.

Contents

Abstract v

Summary vii

Acknowlegdements ix

Contents xi

List of Figures xv

List of Tables xvii

List of Algorithms xix

Part I Introduction 1

1 Overview 3
- 1.1 Motivation . . . 3
- 1.2 Main Contributions of this Thesis . . . 6
- 1.3 Thesis Outline . . . 6

2 Mathematical Preliminaries 9
- 2.1 Convex Analysis . . . 9
- 2.2 Duality . . . 10
- 2.3 Variational Calculus and Total Variation . . . 11

3 Minimal Surface Reconstruction via Binary Segmentation 15
- 3.1 Minimal Surfaces . . . 15
- 3.2 Geometric Properties of the Total Variation . . . 16
- 3.3 Minimal Surfaces for 3D Reconstruction . . . 17

4 Nonsmooth Convex Optimization 19
- 4.1 Convex Relaxation . . . 19
- 4.2 Properties of Optimization Problems . . . 22
  - 4.2.1 Constrainted Optimization Problems . . . 22
  - 4.2.2 Saddle Point Problems . . . 23
  - 4.2.3 Extremality Conditions . . . 24
- 4.3 Algorithms for Total Variation Minimization . . . 24
  - 4.3.1 Gradient Descent . . . 25
  - 4.3.2 Lagged Diffusivity Fixed Point Iterations . . . 26
  - 4.3.3 Fast Iterative Shrinkage and Thresholding Algorithm . . . 27
  - 4.3.4 First Order Primal-Dual Algorithm . . . 28

4.3.5 Convergence Criteria . . . 31
4.4 Discretization . . . 31

Part II Single-View Reconstruction 35

5 Introduction 37
5.1 Related Work and Classification of Single-View Reconstruction Algorithms . 38
5.1.1 Image Cues . . . 38
5.1.2 Priors . . . 40
5.2 Classification of Single-View Approaches . . . 41
5.2.1 Curved Objects . . . 42
5.2.2 Piecewise Planar Objects and Scenes . . . 44
5.2.3 Learning Specific Objects . . . 45
5.2.4 3D Impression from Scenes . . . 45
5.3 Properties and Comparison of Related Works . . . 46
5.4 Problem Setting and Approach . . . 48
5.4.1 Problem Statement . . . 49
5.4.2 Our Approach to Single-View Reconstruction . . . 49
5.4.3 Workflow of Our Approach . . . 50
5.5 Conclusion . . . 50

6 Single-View 3D Reconstruction with a Shape Prior 51
6.1 Introduction . . . 51
6.2 Variational Framework for Single-View Reconstruction . . . 52
6.2.1 Variational Formulation . . . 52
6.2.2 Silhouette Consistency . . . 53
6.2.3 Volume Inflation . . . 53
6.2.4 Optimization via Convex Relaxation . . . 54
6.3 Interactive Single-View Reconstruction . . . 54
6.3.1 Interactive Editing . . . 54
6.3.2 Implementation . . . 56
6.4 Experiments . . . 56
6.5 Conclusion . . . 58

7 Single-View 3D Reconstruction with a Volume Prior 59
7.1 Introduction . . . 59
7.2 Fixed-Volume Minimal Surface Formulation . . . 60
7.2.1 Volume Constraint . . . 60
7.2.2 Fast Minimization . . . 61
7.2.3 Optimality Bounds . . . 63
7.3 Theoretical Analysis of Material Concentration . . . 63
7.4 Experimental Results . . . 64
7.4.1 Cheeger Sets and Single-View Reconstruction . . . 64
7.4.2 Fixed Volume vs. Shape Prior . . . 65
7.4.3 Varying the Volume . . . 65
7.4.4 Weighted Minimal Surface Reconstruction . . . 66
7.5 Conclusion . . . 67

8 Single-View 2.5D Reconstruction with a Volume Prior 69
8.1 Introduction . . . 69
8.2 Fixed Volume Minimal Surfaces on a Two-Dimensional Grid . . . 70

8.3 Minimization of the Proposed Energy . . . 71
8.3.1 Numerical Optimization . . . 72
8.3.2 Implementation . . . 73
8.3.3 Weighted Minimal Surfaces . . . 74
8.4 Experimental Results . . . 74
8.4.1 Qualitative Comparison to Related Methods . . . 74
8.4.2 Experimental Evaluation of our Approach . . . 75
8.5 Conclusion . . . 78

9 Comparison of Approaches **81**
9.1 Comparison of Approaches for Curved Surface Reconstruction . . . 81
9.1.1 Theoretical Comparison . . . 81
9.1.2 Experimental Comparison . . . 82
9.2 Conclusion . . . 90

Part III Spatio-Temporal Multi-View Reconstruction **91**

10 Introduction **93**
10.1 Problem Statement and Notation . . . 94
10.2 Related Work . . . 96
10.2.1 Related Work on Multi-view Stereo Reconstruction . . . 96
10.2.2 Related Work on Spatio-temporal Multi-view Stereo Reconstruction . . . 97

11 Spatio-Temporal Multi-View 3D Reconstruction **101**
11.1 Introduction . . . 101
11.2 Variational Space-Time Reconstruction . . . 102
11.2.1 Photoconsistency Estimation . . . 103
11.2.2 Data Term for Multi-View Reconstruction . . . 104
11.3 Global Optimization . . . 106
11.4 Implementation . . . 106
11.5 Results . . . 107
11.5.1 Photoconsistency and Data Term Evaluation . . . 107
11.5.2 Temporal Regularization . . . 109
11.6 Conclusion . . . 110

12 Surface Normal Integration and Spatially Anisotropic Regularization **113**
12.1 Introduction . . . 113
12.1.1 Related Work . . . 114
12.1.2 Contributions . . . 114
12.2 Variational Space-Time Reconstruction Model . . . 115
12.3 Surface Normal Integration . . . 116
12.3.1 Photoconsistency and Data Term Estimation . . . 116
12.3.2 Normal Estimation . . . 118
12.4 Optimization . . . 118
12.5 Implementation . . . 120
12.6 Results . . . 121
12.7 Conclusion . . . 124

13 Generalized Connectivity Constraints **125**
13.1 Introduction . . . 125
13.1.1 Contributions . . . 126

13.1.2 Related Work on Connectivity Constraints . . . 126
13.2 Review of Connectivity Constraints for Image Segmentation . . . 127
13.3 3D Reconstruction with Connectivity Constraints . . . 129
13.4 Generalized Connectivity Constraints for Objects of Arbitrary Genus . . . 130
13.4.1 Handle and Tunnel Loops . . . 130
13.4.2 Loop Connectivity Constraints . . . 132
13.5 Numerical Optimization . . . 133
13.6 Experiments . . . 135
13.7 Conclusion . . . 136

Part IV Conclusions and Outlook **139**

14 Concluding Remarks **141**

15 Limitations and Future work **143**
15.1 Single-View Reconstruction . . . 143
15.2 Spatio-Temporal Multi-View Reconstruction . . . 143

Notation **147**

Own Publications **149**

References **151**

List of Figures

1.1 Multi-view 3D reconstruction illustration . . . 3

2.1 Illustration of convex, non-convex functions, the epigraph, the convex conjugate, and the convex envelope. . . . 11
2.2 Illustration of total variation. . . . 12

3.1 Illustration of minimal surfaces formed by soap bubbles. . . . 15
3.2 Illustration of the indicator function $\mathbf{1}_S(\boldsymbol{x})$. . . . 16

4.1 Nonsmooth convex optimization illustration. . . . 19
4.2 Illustration of the energy bound. . . . 21
4.3 Illustration of a saddle point problem. . . . 23
4.4 Gradient descent steps on a quadratic function with and without constraints on the feasible domain. . . . 25
4.5 Successive Over-relaxation. Illustration of the linear extrapolation. . . . 27
4.6 Schematic plots of gradient descent iterations. . . . 30

5.1 Illustration of single-view reconstruction. . . . 37
5.2 General workflow of our single-view reconstruction approach. . . . 50

6.1 Input images and textured reconstruction results from the method proposed in this chapter. . . . 51
6.2 Illustration of our volumetric setup and notation. . . . 52
6.3 Illustration of the data term consisting of silhouette constraints and our proposed distance-based shape prior. . . . 53
6.4 Parameters affecting the shape priors. . . . 55
6.5 Visualization of different shape priors and the influence of user input on the reconstruction result. . . . 55
6.6 Input images (1st column) and corresponding reconstruction results (2nd-4th column): textured model, untextured geometry, textured model without image plane. . . . 57
6.7 Possible applications of our single-view reconstruction approach. Novel view synthesis and change of material and reflectance properties of the surface. . . . 58

7.1 Single-view 3D reconstruction results with a volume prior. . . . 59
7.2 Illustration of the projection scheme by Boyle and Dykstra [31]. . . . 62
7.3 The two cases considered in the analysis of the material concentration. . . . 63
7.4 The inflation of the reconstruction model can be intuitively changed by varying the target volume. . . . 64
7.5 The proposed volume prior approach favors minimal surfaces for a user-specified volume. Therefore the reconstruction algorithm is ideally suited to compute smooth, round reconstructions. . . . 65
7.6 Effects of the volume prior and the shape prior in comparison. . . . 65
7.7 Comparison of results using the volume prior vs. shape prior. . . . 66

7.8 Generating sharp edges in 3D models via user-defined regularization weights. 66
7.9 Volume inflation dominates where the silhouette area is large (bird) whereas thin structures (twigs) are inflated less. 67
7.10 Output comparison of single-view methods with shape and volume prior. . . . 68

8.1 Reconstruction result with the proposed single-view 2.5D reconstruction approach with a volume prior. 69
8.2 The area of an infinitesimal surface element based on partial derivatives. . . . 70
8.3 Runtime comparison of different algorithms minimizing Equation (8.4) measured on the teapot example without user-scribbles. 76
8.4 Comparison of reconstruction results for several single-view methods. 77
8.5 Comparison of the 2.5D approach vs. the 3D volumetric approach. 78
8.6 Influence of the volume parameter on the reconstruction. 78
8.7 Reconstruction results with user input altering the local smoothness of the surface. 79

9.1 Experimental comparison of several methods for curved object reconstruction. 83
9.2 Continuation of Figure 9.1: 84
9.3 Continuation of Figure 9.1: 85
9.4 User input for the methods of Zhang et al. [253] and Igarashi et al. [117]. . . 87
9.5 Overview of necessary and optional steps and user input for Prasad et al. [183]. 87

10.1 Spatio-temporal Multi-view 3D reconstruction overview. 93
10.2 Visual Hull as the intersection of silhouette pre-images (picture from [25]). . . 95

11.1 Space time surface evolution. 101
11.2 Outline of the proposed space time reconstruction framework. 102
11.3 Schematic plots of probabilities along a camera ray. 105
11.4 Comparison of the data term from Kolev et al. [135] (a) and the proposed one (b) for a lower cross section of the skirt. 107
11.5 Comparison of the reconstruction results using the data term by Kolev et al. [135] and the proposed one. 108
11.6 Comparison of the proposed method for $|T| = 1$ with other 3D reconstruction methods. 109
11.8 Illustration of the exponential temporal weighting. 109
11.7 Results of our framework on several data sets for $|T| = 3$. 110
11.9 Effect of the temporal regularization. 111
11.10 Comparison of different reconstruction techniques. 111

12.1 Reconstruction results for normal integration and anisotropic regularization. . 113
12.2 Effects of the proposed normal integration. 122
12.3 Comparison of our results to other methods 122
12.4 Reconstruction results on different scenes (rope jump, boy cartwheel, stick) from [121]. 124

13.1 The effect of connectivity constraints and their generalization to higher topology in comparison. 125
13.2 Illustration of the connectivity constraints for image segmentation. 128
13.3 Visualization of various sets on a teapot model of genus 2. 131
13.4 Comparison of the two connectivity constraints. 133
13.5 Visualization of various visual hull properties. 134
13.6 Comparison of results to other state-of-the-art approaches. 137

List of Tables

5.1 Overview and classification of single-view methods. 47

8.1 Runtime comparison of the 3D approach vs. the 2.5D approach with volume prior. 76

9.1 Comparison of approximate modeling times for different single-view reconstruction methods. 86
9.2 Necessary and optional user inputs and modeling steps for several single-view methods in comparison. 88
9.3 Overview of advantages and disadvantages of single-view reconstruction methods. 89

11.1 Average runtimes per frame for our spatio-temporal 3D reconstruction method on different data sets. 109

List of Algorithms

1 Projected Gradient Descent (PGD) 25
2 Lagged Diffusivity Fixed Point Iterations (LDFPI) 26
3 Successive Over-Relaxation (SOR) 27
4 Fast Iterative Shrinkage and Thresholding Algorithm (FISTA) 28
5 First Order Primal-Dual (PD) Algorithm 29
6 Preconditioned First Order Primal-Dual (PD) Algorithm 30

Part I.

Introduction

1. Overview

Insight must precede application.

Max Planck
(1858-1947)

1.1. Motivation

Humans navigate in the world mostly based on what they see with their eyes. A long standing goal and a core area of research in Computer Vision is to enable computers to perform similar complex tasks based on images acquired with cameras. Apart from autonomous navigation, the need to simply and cheaply obtain accurate 3D models of the world arose in many areas of research and industry, and is further growing, for instance, in architectural planning, civil engineering, medical imaging, cultural heritage archiving, as well as for the movie and entertainment industry.

The basic task is to get an understanding of the 3D world from the 2D data available in form of images. These images arise from the camera sensors measuring the amount of incoming light that has been reflected by surrounding objects. Naturally, the depth information, that is, distances between objects and the camera are not measured and are therefore lost in the acquisition process. The goal of 3D reconstruction is to recover the depth and geometry of the scene from a given set of images. This is called an *inverse problem*, because one tries to invert the image formation process in the camera. If several images of the same scene from different view points are available, 3D reconstruction can be performed by solving a

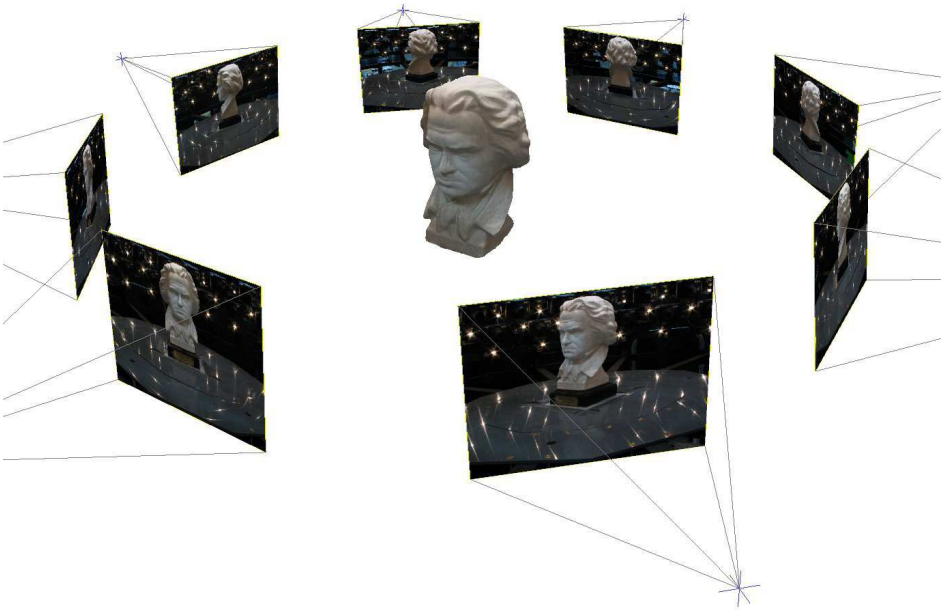


Figure 1.1.: Multi-view 3D reconstruction of a Beethoven bust. Given a set of images which observe the object from different view points, the goal is to recover the 3D geometry of the object.

correspondence problem, that is, estimating which 2D points in the images belong to the same point in the 3D world. Figure 1.1 illustrates a 3D reconstruction setup from multiple input images.

This multi-view 3D reconstruction problem is still very difficult, because the input data is usually noisy, or insufficient, if parts of the scene are unobserved or occlude each other in the images. Further, the matching problem can be extremely difficult. Depending on the objects' structure, material properties and lighting conditions, parts of the same object can appear very differently from different view points, for example, due to specular reflections on shiny surfaces.

Nowadays, a variety of combined image and depth sensors are available, which increasingly have useful output resolutions and affordable prices. Therefore, a legitimate question to ask is why inferring depth information from 2D images is still a problem that deserves research interest. There are several arguments to encounter that question: 1) Several depth cameras rely on active sensors (e.g. structured light or time-of-flight cameras) which, in contrast to classical cameras, have limited range or do not work for arbitrary light conditions. 2) Currently, all depth cameras are considerably larger than classical cameras which also limits their applicability (e.g. for endoscopy in medical imaging). 3) Generally, 3D reconstruction from images is necessary when depth sensors are not applicable or available (e.g. recovering geometry from historic images). 4) Depth sensors are still and will probably remain more expensive than classical cameras. Therefore, 3D reconstruction from images is still an important problem to solve.

For decades the 3D reconstruction problem has been receiving a large amount of attention in research and a wide variety of methods and corresponding literature exists. Many of these works have studied the problem of two cameras observing a static scene, and extensions to non-static scenes exist too. Another significant number of these works deal with a slightly more general problem of having an arbitrary number of cameras observing a static scene. In contrast to the two-camera stereo reconstruction approaches, which usually compute a depth map with respect to one of the cameras, a general multiple camera setup can be used to obtain a full and dense 3D model of the scene. Nevertheless, there are more cases of 3D reconstruction that can be considered.

In this thesis we investigate two extreme cases of the 3D reconstruction problem: 1. The reconstruction of objects from only a single image which we will refer to as *single-view reconstruction*, and 2. the reconstruction of a dynamic scene from multiple, synchronously captured input videos which we will call *spatio-temporal multi-view reconstruction*.

Single-View Reconstruction. For the single-view case the inverse problem is much harder, because a correspondence with other views cannot be attained. To deal with this additional complexity, most approaches try to solve a simplified problem by restricting the type of input images, the class of scenes or objects, their material properties or they take the human in the loop to assist the reconstruction. Similar to most other works in this field our approach will be a combination of the aforementioned simplifications, that is, we will focus on a small but reasonable subclass of objects and most importantly will only require a very small amount of human interaction to obtain plausible 3D reconstructions from a single input image.

Spatio-Temporal Multi-View Reconstruction. In the multi-view case the problem is almost contrary. Instead of rather little image information one has to deal with a very large amount of input data and the problem how to use this data efficiently and how to get the most out of it. Compared to the static scene case, the point correspondence problem is significantly harder, because objects in the scene might move or even deform over time. A reasonable question

to ask is why should one look for temporal correspondences in the first place. Why is it not sufficient to use the technology we already have and perform an ordinary 3D reconstruction at every time step independently? The short answer is that the output quality of any 3D reconstruction algorithm is bound to the resolution and quality of input images and the use of additional image data from other time steps can result in an improved reconstruction accuracy. Similar ideas have also been used for image enhancing techniques from several input images, for example super-resolution [220]. A more detailed discussion of the motivation, challenges and benefits of adding temporal coherence will be given in Part III.

One goal of this thesis is to work towards a unified general 3D reconstruction model that is easily extendible and applicable to a variety 3D reconstruction scenarios. Another important goal is to formulate appropriate priors that help to deal with the ill-posedness of the reconstruction task. Psychologists [151, 172] have identified several priors in the human visual system that serve as heuristics to interpret what we see and do not see, namely: symmetry, planarity, maximum compactness, and minimum surface. A big challenge is to integrate these priors into a practicable 3D reconstruction approach. In this thesis we will look at some of these priors and demonstrate their usefulness in various scenarios.

The most popular and practicable prior is the minimum surface prior, because it successfully deals with most of the common problems in 3D reconstruction: missing data, redundant data, measurement noise and outliers.

In this thesis we will adopt a variational 3D reconstruction approach which allows to compute a minimal surface as a critical point of an energy cost function. In Part II, we propose several priors and an alternative surface representation to tailor the reconstruction framework for the case of single-view reconstruction. In Part III, we generalize the variational 3D reconstruction approach to the spatio-temporal multi-view case, for which we also propose a novel data term, an extension with surface normal estimation and a novel generalized connectivity prior.

1.2. Main Contributions of this Thesis

The main contributions of this thesis can be summarized in the following points:

- **Convex variational formulation of single-view reconstruction.** To the best of our knowledge we present the first variational approach to single-view reconstruction based on convex energy minimization that allows to reconstruct objects of arbitrary topology. This approach in combination with different priors and scene representations is described in Part II, Chapters 5 to 9, and has been presented in several publications [1, 2, 3, 4, 5, 6].
- **Convex variational formulation of spatio-temporal multi-view reconstruction.** To the best of our knowledge we present the first variational spatio-temporal multi-view reconstruction based on convex energy minimization. This approach and two extensions are discussed in Part III, Chapters 10 to 13, and have been published in [7, 8, 9].
- **Efficient topological constraints for 3D reconstruction.** To the best of our knowledge we present the first work on multi-view 3D reconstruction in which constraints on the topological genus (i.e. the number of holes of an object) can be efficiently enforced. In particular we can guarantee that the topological genus of the reconstruction is not smaller than the one of the visual hull. This work is presented in Chapter 13 and has been published in [9].
- **Significant extensions toward a general and unified model for 3D reconstruction.** The basis of this thesis is the variational 3D reconstruction approach by Kolev et al. [134]. We have tailored this approach for its use in two very different 3D reconstruction sub-problems by proposing and adding a variety of novel, useful priors and constraints. In fact, the spatio-temporal multi-view approach in Part III is a true generalization as it reduces to the approach by Kolev et al. [134] if the scene is static or only one time instant is considered. The surface representation, the efficient computation via convex energy minimization, and the extensibility with a variety of priors, makes the approach suitable for many 3D reconstruction scenarios - as demonstrated in this thesis. To the best of our knowledge no other approach in the literature has been shown to work well on such a variety of 3D reconstruction problems.

1.3. Thesis Outline

This thesis is organized in 15 chapters which are grouped into the following four parts:

- Part I: Introduction (Chapters 1 to 4)
- Part II: Single-View Reconstruction (Chapters 5 to 9)
- Part III: Spatio-Temporal Multi-View Reconstruction (Chapters 10 to 13)
- Part IV: Conclusions and Outlook (Chapters 14 and 15)

After a general introduction into the problem of recovering geometry from images, each part will refine the problem statement to its specific conditions. Nevertheless, we will use the same general reconstruction framework for both input scenarios and investigate several variants for each scenario. The particular contents of each part are detailed in the following paragraphs.

Part I: Introduction. After an overview in **Chapter 1** which motivates and outlines this thesis, **Chapter 2** introduces the mathematical background on convex analysis, duality theory, variational calculus and the total variation norm which form the basis of all surface

reconstruction approaches in this thesis.

Chapter 3 explains the surface representation which is based on the geometric properties of the total variation to describe minimal surfaces. This leads to a general model for 3D reconstruction of which several variants are studied in Part II for the single-view reconstruction case and which is extended to the spatio-temporal multi-view reconstruction scenario in Part III.

Chapter 4 discusses how these surfaces can be efficiently computed by transforming the optimization problems into equivalent ones which are easier to solve. After describing necessary conditions for computing minima, we present several numerical algorithms for non-smooth convex optimization problems which are suitable for efficiently solving these minimal surface problems. We further explain numerical details such as the proper discretization of the derivate operators.

Part II: Single-View Reconstruction. **Chapter 5** gives an introduction to the single-view reconstruction problem and provides an overview on related work. Moreover, we propose a taxonomy to classify different single-view reconstruction approaches based on several properties, such as the scenes representation, considered object classes, reconstruction accuracy, etc. The literature overview and classification is part of a survey paper published in [6].

Chapter 6 introduces the first variational framework for convex single-view reconstruction using a shape prior. This work has been published at DAGM 2009 [1] and also appeared as part of book chapters [4, 2, 6]. The work has been awarded the DAGM paper prize.

Chapter 7 discusses a variant of the single-view framework in which the shape prior is replaced by a volume prior. This tackles several shortcomings of the shape prior approach and further reduces the amount of necessary user input. This chapter is based on work published at ACCV 2010 [3] and a comparison to the shape prior approach appeared in the book chapters [4, 2, 6]. This work received an ACCV honorable mention award.

Chapter 8 presents another variant of the single-view framework with a volume prior. We show that the same problem can be solved more efficiently and accurately by replacing the memory intensive implicit surface representation with a simpler height field representation. This chapter contains work published at CVPR 2012 [5].

Chapter 9 provides a thorough comparison of our single-view approaches with shape and volume priors to the most related state-of-the-art approaches. This comparison is also part of the survey paper published in [6].

All published works in this part have been conducted in close collaboration with Eno Töppe and have also been part of his PhD thesis [211].

Part III: Spatio-Temporal Multi-View Reconstruction. **Chapter 10** gives an introduction to spatio-temporal multi-view reconstruction, explains challenges and provides an overview of related work.

Chapter 11 introduces the first variational framework for convex spatio-temporal multi-view reconstruction. We further propose a novel data term that is better suited for sparse camera setups. Moreover, we propose to compute photoconsistency matches differently and with lower complexity than previous approaches, which lead to significantly lower computation times with a similar reconstruction quality. This framework has been presented at the ICCV 4DMOD workshop [7].

Chapter 12 proposes several improvements of the spatio-temporal multi-view reconstruction approach. In particular, we propose to estimate surface normals in an iterative reconstruction

approach and demonstrate that these surface normals can be used to 1) improve photometric matching scores, and 2) regularize the surface in an anisotropic manner that better preserves small surface details. This chapter presents work published at BMVC 2014 [8].

Chapter 13 demonstrates how recent advances on imposing connectivity constraints can be 1) integrated into our spatio-temporal multi-view reconstruction framework, and 2) generalized to useful topological constraints in order to maintain the connectivity of objects with arbitrary topological genus. This work has been published at ECCV 2014 [9].

Part IV: Conclusions and Outlook. The last part of the thesis summarizes the achievements of this work in **Chapter 14**. Finally, in **Chapter 15**, we discuss shortcomings of the proposed approaches together with possible directions for future work.

2. Mathematical Preliminaries

Perplexity is the beginning of knowledge.

Khalil Gibran
(Lebanese Poet, 1883 - 1931)

This chapter and the following two chapters define the mathematical framework for this thesis by introducing basic concepts and notations for describing and computing images and surfaces in two, three and four dimensional spaces.

2.1. Convex Analysis

In this section we introduce several properties of functions that will be necessary to efficiently compute minimal surfaces as a minimizer of an objective function. The following definitions and properties are basic definitions and can be found widely in the respective literature, see for instance [187, 188, 26]. An important and strong property for sets and functions is the notion of convexity. In particular for functions, this property helps to proof the existence and uniqueness of minimizers and simplifies the optimization of such functions drastically, which will also be discussed in Chapter 4.

Definition 2.1 (Convex Set)**.** *A set C is called convex if $\lambda x + (1-\lambda)y \in C$ for all $x, y \in C$ and $0 \leq \lambda \leq 1$.*

Geometrically this means that a set is convex if and only if the connecting line between any two points in the set is also entirely contained in the set.

Convexity is not only a property of sets but also for functions defined over a convex domain. Considering a function $f : \mathcal{V} \to \mathbb{R}$ with domain $\mathcal{V}$, the graph of the function being defined as $\{(x, f(x)) \mid x \in \mathcal{V}\}$ divides the function space into two sets which are (1) all points above the graph - called *epigraph* - and (2) all points below the graph - called the *hypograph*.

Definition 2.2 (Epigraph)**.** *The epigraph of a function $f : \mathcal{V} \to \mathbb{R}$ is the set of all points which lie above or on the graph:*

$$\operatorname{epi} f = \{(x, t) \mid x \in \mathcal{V}, t \in \mathbb{R}, f(x) \leq t\} \quad . \tag{2.1}$$

The epigraph links the convexity property of functions with the one of sets. A function f is then said to be convex if and only if the epigraph of the function is a convex set. This also implies that the function domain needs to be a convex set. However, in the literature one usually finds the following equivalent definition which is often easier to verify on more regular functions.

Definition 2.3 (Convex Function)**.** *A function $f : \mathcal{V} \to \mathbb{R}$ is called convex if the function domain $\mathcal{V}$ is a convex set and if*

$$f(\lambda x + (1-\lambda)y) \;\leq\; \lambda f(x) + (1-\lambda) f(y) \qquad \forall x, y \in \mathcal{V} \text{ and } 0 \leq \lambda \leq 1 \;. \tag{2.2}$$

Function f is called *strictly convex* if the inequality in Equation (2.2) holds strictly for all $x, y \in \mathcal{V}$.

Further, a function f is called (strictly) *concave* if the function $-f$ is (strictly) convex.

The visual meaning of Equation (2.2) is the following: For any two points $(x, f(x))$ and $(y, f(y))$ on the graph of f, each point on the line segment that connects the points must lie above the graph (or on the graph for non-strict convexity). See Figure 2.1(a) for an illustration of a convex function and Figure 2.1(b) for the non-convex case.

The definition of convex functions in Equation (2.2) is called the *zero-order condition* for describing convex functions. For differentiable functions f, convexity is equivalent to *first-order condition*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \qquad \forall x, y \in \mathcal{V} \ , \tag{2.3}$$

which means that the function f is globally above the tangent at x. If f is even twice-differentiable an equivalent *second-order condition* is the positive semi-definiteness of the Hessian:

$$\nabla^2 f(x) \succeq 0 \qquad \forall x \in \mathcal{V} \ , \tag{2.4}$$

which means that the function f is either flat or curved upwards in every direction.

The following function properties will occur occasionally as technical requirements in definitions and theorems.

Definition 2.4 (Proper Convex Function)**.** *A convex function $f : \mathcal{V} \to \mathbb{R} \cup \{-\infty, \infty\}$ is called proper if its epigraph is non-empty and contains no vertical lines, or equivalently, if $f(x) < +\infty$ for at least one x and $f(x) > -\infty$ for all x.*

Many properties and proofs in convex analysis require functions to be continuous, however, often it is sufficient to assume the following weaker property:

Definition 2.5 (Lower Semi-Continuous Function)**.** *A function $f : \mathcal{V} \to \mathbb{R}$ is lower semi-continuous if and only if its epigraph is closed, or equivalently, if it is lower semi-continuous at every point $x \in \mathcal{V}$. The function $f(x)$ is lower semi-continuous at point x, if*

$$f(x) = \liminf_{y \to x} f(y) = \lim_{\epsilon \downarrow 0} (\inf\{f(y) \mid |x - y| \leq \epsilon\}) \ . \tag{2.5}$$

The combination of lower semi-continuity and, its analogue, upper semi-continuity yields ordinary continuity.

2.2. Duality

Definition 2.6 (Convex Conjugate - Legendre-Fenchel Transform)**.** *Let $f : \mathcal{V} \to \mathbb{R}$ be a function. Then, the function $f^* : \mathcal{V}^* \to \mathbb{R}$,*

$$f^*(\boldsymbol{p}) = \sup_{\boldsymbol{x} \in \mathcal{V}} \{\langle \boldsymbol{p}, \boldsymbol{x} \rangle - f(\boldsymbol{x})\} \tag{2.6}$$

is called the convex conjugate or Legendre-Fenchel transform of the function f. $\mathcal{V}^*$ is called the dual space of $\mathcal{V}$.

Some vector and function spaces are self-dual, for instance for $\mathcal{V} = \mathbb{R}^n$, it holds that $\mathcal{V} = \mathcal{V}^*$

The idea behind the Legendre-Fenchel transform is to represent the function f in the space of supporting lines (or hyperplanes) of the graph being represented as tuples of the slope (or

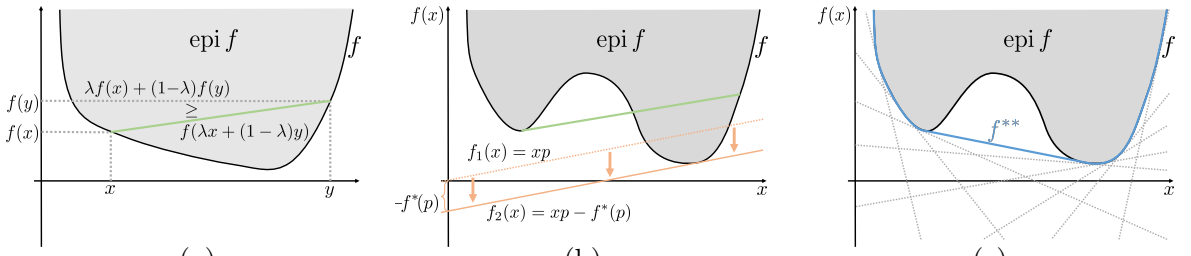


Figure 2.1.: Illustration of convex, non-convex functions, the epigraph, the convex conjugate, and the convex envelope. (a) shows the graph of the function (black), the epigraph (gray) is the set of points above the function graph. For convex functions the line segment (green) between two arbitrary points on the graph should be entirely above the graph or touching it. (b) Function graph of a non-convex function. The green line segment is partially below the graph. The orange line illustrates the computation of the convex conjugate. For any given slope p a graph-supporting line is computed which has intercept $-f^*(p)$. (c) Considering all possible slopes p, the set of all supporting lines forms the convex envelope f^{**}.

plane normal) and the corresponding maximal intercept. Hence, the *supremum* operation is used for the transformation from the space of $(\boldsymbol{x}, f(\boldsymbol{x}))$ to the space of gradient and conjugate $(\boldsymbol{p}, f^*(\boldsymbol{p}))$.

The *convex biconjugate* $f^{**} = (f^*)^* \leq f$ is the maximal convex function below f and represents the convex hull of the epigraph of f. It is also called the *convex envelope* of function f. See Figure 2.1(b,c) for an illustration of these definitions.

Theorem 2.7 (Fenchel-Moreau)**.** *For proper convex, lower semi-continuous functions f, $f = f^{**}$ holds true.*

Proofs of this theorem can be found in [187, 44] or [188, page 474]. Another important result of duality theory is the concept of the adjoint operator (see e.g. [33, Def. 2.23]).

Definition 2.8 (Adjoint Operator)**.** *Let $\mathcal{H}_x, \mathcal{H}_y$ be Hilbert spaces, with respective inner products $\langle\cdot,\cdot\rangle_{\mathcal{H}_x}$, $\langle\cdot,\cdot\rangle_{\mathcal{H}_y}$ and let $A : \mathcal{H}_x \to \mathcal{H}_y$ be a continuous linear operator. One can show that there exists a unique continuous linear operator $A^* : \mathcal{H}_y \to \mathcal{H}_x$ having the following property:*

$$\langle Ax, y\rangle_{\mathcal{H}_y} = \langle x, A^*y\rangle_{\mathcal{H}_x} \qquad \forall x \in \mathcal{H}_x, y \in \mathcal{H}_y. \tag{2.7}$$

Operator A^* is called the adjoint operator of A.

Together with the concept of the *weak derivative* which will be described in the next section, the adjoint operator will be useful to "shift" differential operators from one variable to other within a scalar product. Further, the property that two operators are *adjoint* will be necessary when discretizing differential operators for duality-based numerical solvers.

2.3. Variational Calculus and Total Variation

So far, we did not specify the function domain $\mathcal{V}$ in the definitions above, which can for example be finite, e.g. a subset of $\mathbb{R}^n$. However, we will also consider *infinite* function domains, because in this thesis we want to recover surfaces which are described by functions. In order to evaluate the quality of different surface reconstructions we look at energy functions that assign a cost to each surface, that is, a function which takes a function as its argument and returns a real number. Mathematically, this is called a *functional* and is the basis of an entire field in mathematics, the *calculus of variations*, which studies their properties.

Definition 2.9 (Functional)**.** *A functional $E : \mathcal{V} \to \mathbb{R}$ is a real-valued function on a vector space $\mathcal{V}$ which assigns every element of the space a real number.*

In our setup the vector space $\mathcal{V}$ will be a space of functions, that is, $\mathcal{V} = \{u : \mathbb{R}^n \to \mathbb{R}\}$. Note that all previous definitions hold for functions as well as for functionals, although some operators such as the gradient need to be generalized, but we will only discuss the details we need (see [13] for more details).

The following functional will play a central role in this thesis, because it defines a measure of "smoothness" for a given function and has many useful properties which are discussed in the next chapter.

Definition 2.10 (Total Variation for Differentiable Functions)**.** *Let $u \in \mathcal{C}^1(\Omega, \mathbb{R}^n)$ be a differentiable function, then*

$$\mathrm{TV}(u, \Omega) = \int_\Omega |\nabla u|_2 \, d\boldsymbol{x} \tag{2.8}$$

is called the total variation of u on domain Ω.

The set $\mathcal{C}^k(\Omega, \mathbb{R}^n)$ denotes the set of functions $f : \Omega \to \mathbb{R}^n$ being k-times differentiable. For better readability and completeness we will define all function spaces only at the end of this section.

The total variation sums up all absolute height differences of a function. If we interpret a function as an elevation profile of a hiking trail, the total variation only measures the sum of all altitude differences for going up and down the hill - regardless of how long the trail is. In that sense, the length of the trail or its steepness is irrelevant as long as it adds up to the same height difference. A climber who climbs straight up a vertical wall has the same effort (total variation) as the hiker who takes a longer walking trail to get to the same peak. See Figure 2.2 for an illustration.

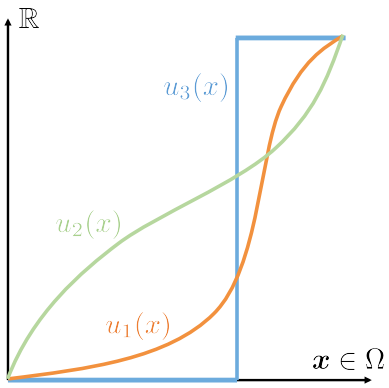


Figure 2.2.: Illustration of total variation. All these functions have the same total variation on the depicted interval.

For many practical purposes this definition of the total variation can also be generalized for non-differentiable functions using the concept of weak derivates. Motivated by the "integration by parts" technique one can define a weak derivative for functions which are not everywhere differentiable in the classical sense (i.e. they do not have a *strong* derivative).

Definition 2.11 (Weak Derivative)**.** *Let $\Omega \subset \mathbb{R}^n$ and $u \in \mathcal{L}^1(\Omega)$, then function $v \in \mathcal{L}^1(\Omega)$ is a weak derivative of u if,*

$$\int_\Omega u \cdot \mathrm{div}(\boldsymbol{p}) \, d\boldsymbol{x} = -\int_\Omega v \cdot \boldsymbol{p} \, d\boldsymbol{x} \tag{2.9}$$

for all functions $\boldsymbol{p}$ being infinitely differentiable and with compact support in Ω, i.e. $\boldsymbol{p} \in \mathcal{C}_c^\infty(\Omega, \mathbb{R}^n)$.

This relationship essentially represents the integration by parts formula, because the third integral over the boundary of Ω vanishes due to the fact that $\boldsymbol{p}$ and all its derivatives are zero on the boundary. The principle idea behind the weak derivative is that Equation (2.9) allows to shift the differential operator from one variable to other which is defined to be always differentiable. As long as the two integrals in Equation (2.9) sum up to the same value v is a weak derivative of u and everywhere where u has a classical derivative it holds

that $v(\boldsymbol{x}) = \nabla u(\boldsymbol{x})$. Note that the integral in Equation (2.9) corresponds to an inner product in the space $\mathcal{L}^1(\Omega)$, that is, $\langle u, \operatorname{div}(\boldsymbol{p})\rangle = \int_\Omega u \cdot \operatorname{div}(\boldsymbol{p})\, d\boldsymbol{x}$. Then, according to the definition of the adjoint operator (Definition 2.8) the symbols $\operatorname{div}(\boldsymbol{p})$ and $-\nabla u$ in Equation (2.9) are adjoint (for $v = \nabla u$).

Further, consider the following property of the 2-norm for non-zero arguments

$$|\nabla u|_2 = |\nabla u|_2 \cdot \frac{|\nabla u|_2}{|\nabla u|_2} = \frac{\langle \nabla u, \nabla u\rangle}{|\nabla u|_2} = \langle \nabla u, \underbrace{\frac{\nabla u}{|\nabla u|_2}}_{\boldsymbol{p}}\rangle \quad \text{for } \nabla u \neq 0 \ , \tag{2.10}$$

which leads to a *dual representation* of the 2-norm

$$|\nabla u(\boldsymbol{x})|_2 = \sup_{\|\boldsymbol{p}(\boldsymbol{x})\|_2 \leq 1} \langle \nabla u(\boldsymbol{x}), \boldsymbol{p}(\boldsymbol{x})\rangle \ . \tag{2.11}$$

As Chan et al. [48] proposed, we can use the definition of the weak derivative (Definition 2.11) and the dual representation of the 2-norm in Equation (2.11) to derive a more general definition of the total variation for weakly differentiable functions:

Definition 2.12 (Total Variation (TV))**.** *A functional of the form*

$$\mathrm{TV}(u, \Omega) := \sup \left\{ -\int_\Omega u \cdot \operatorname{div}(\boldsymbol{p})\, d\boldsymbol{x} \;\middle|\; \boldsymbol{p} \in \mathcal{C}_c^1(\Omega, \mathbb{R}^n), \|\boldsymbol{p}\|_{\mathcal{L}^\infty(\Omega)} \leq 1 \right\} \tag{2.12}$$

is called total variation or variation of u on domain Ω.

In fact, Definition 2.12 is a generalization of the total variation in Definition 2.10 to weakly differentiable functions. For differentiable functions $u \in \mathcal{C}^1(\Omega, \mathbb{R})$ Definition 2.10 and Definition 2.12 are equivalent with the vector field $\boldsymbol{p} \in \mathcal{L}^1(\Omega, \mathbb{R}^n)$ defined by

$$\boldsymbol{p}(\boldsymbol{x}) = \begin{cases} \frac{\nabla u(\boldsymbol{x})}{|\nabla u(\boldsymbol{x})|_2} & \text{if } |\nabla u(\boldsymbol{x})|_2 \neq 0 \\ 0 & \text{otherwise} \ , \end{cases} \tag{2.13}$$

Equation (2.12) reduces to Equation (2.8). For ease of notation one commonly denotes the total variation as in Definition 2.12 and remarks that the integral is evaluated in a "distributional sense" as written in Definition 2.10. We will make use of this notation in the rest of this thesis. The dual definition of total variation and related minimization algorithms have been studied extensively in the literature, especially in the context of image restoration, see for example the PhD theses [40, 255, 174] and related publications. Further, Chambolle et al. [44] provide a solid and comprehensive introduction to the theory of total variation and its applications in computer vision.

A simple but useful generalization of the total variation is the *weighted* total variation which has been first studied in [160] and introduced into computer vision by Bresson et al. [34].

Definition 2.13 (Weighted Total Variation)**.** *For a weight function $g : \Omega \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and $u \in \mathcal{L}^1_{\mathrm{loc}}(\Omega, \mathbb{R}^n)$ one defines*

$$\begin{aligned} \mathrm{TV}_g(u, \Omega) &:= \sup \left\{ -\int_\Omega u \cdot \operatorname{div}(\boldsymbol{p})\, d\boldsymbol{x} \;\middle|\; \boldsymbol{p} \in \mathcal{C}_c^1(\Omega, \mathbb{R}^n),\ \forall \boldsymbol{x} \in \Omega :\ \|\boldsymbol{p}(\boldsymbol{x})\|_2 \leq g(\boldsymbol{x}) \right\} \\ &= \int_\Omega g(\boldsymbol{x})\, |\nabla u|_2 \ d\boldsymbol{x} \ , \end{aligned} \tag{2.14}$$

where the second equality only holds for differentiable functions $u \in \mathcal{C}^1(\Omega, \mathbb{R})$.

Since we will make heavy use of the total variation in many objective functionals, an important property for their minimization is its convexity which is not difficult to show.

Proposition 2.14 (Convexity of the Total Variation)**.** *For any function* $u : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ *the functional* $E(u) = \mathrm{TV}(u, \Omega)$ *is convex in* u.

Proof. Using the zero-order convexity condition in Definition 2.3 we select two arbitrary functions $u_1, u_2 : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ and derive

$$\mathrm{TV}(\lambda u_1 + (1-\lambda)u_2) \leq \lambda\,\mathrm{TV}(u_1) + (1-\lambda)\,\mathrm{TV}(u_2) \tag{2.15}$$

$$\Leftrightarrow \quad \int_\Omega |\nabla(\lambda u_1 + (1-\lambda)u_2)|\; d\boldsymbol{x} \leq \lambda \int_\Omega |\nabla u_1|\; d\boldsymbol{x} + (1-\lambda)\int_\Omega |\nabla u_2|\; d\boldsymbol{x} \tag{2.16}$$

$$\Leftrightarrow \quad \int_\Omega |\lambda\nabla u_1 + (1-\lambda)\nabla u_2|\; d\boldsymbol{x} \leq \int_\Omega \Big(|\lambda\nabla u_1| + |(1-\lambda)\nabla u_2| \Big)\; d\boldsymbol{x}\;, \tag{2.17}$$

where the last inequality holds because of the triangle inequality. □

Function spaces. In our notation we will use the following function spaces which group a set of functions according to some property. In particular we will look at functions with finite norms.

Definition 2.15 ($\mathcal{L}^p$ Spaces or Lebesgue Spaces)**.** *A function* $u : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ *is element of the* $\mathcal{L}^p$*-space, written as* $u \in \mathcal{L}^p(\Omega, \mathbb{R})$ *if its corresponding norm is finite*

$$\mathcal{L}^p(\Omega, \mathbb{R}) = \{u \in \mathcal{L}^1_{\mathrm{loc}}(\Omega, \mathbb{R}) \mid \|u\|_p < \infty\} \quad \text{with} \quad \|u\|_p = \left(\int_\Omega |u(\boldsymbol{x})|^p \, d\boldsymbol{x} \right)^{\frac{1}{p}}\;, \tag{2.18}$$

where $\mathcal{L}^1_{\mathrm{loc}}(\Omega, \mathbb{R})$ is the space of locally integrable functions with domain Ω and image $\mathbb{R}$.

Following Definition 2.12 one can define the following function space as a set of functions with bounded (total) variation.

Definition 2.16 (Functions of Bounded Variation (BV-space) [13])**.** *The set of functions with bounded variation, that is, with a variation smaller than infinity, is defined as*

$$\mathcal{BV}(\Omega, \mathbb{R}) = \{u \in \mathcal{L}^1_{\mathrm{loc}}(\Omega, \mathbb{R}) \mid \mathrm{TV}(u, \Omega) < \infty\}\;. \tag{2.19}$$

The total variation defines a semi-norm on the space of bounded variations and is therefore often called TV-norm. Further, we have already made use of the space of differentiable functions $\mathcal{C}^k(\Omega, \mathbb{R}^n)$ being defined as

$$\mathcal{C}^k(\Omega, \mathbb{R}^n) = \{u : \Omega \to \mathbb{R} \mid u \text{ is } k\text{-times continuously differentiable}\}\;. \tag{2.20}$$

With the additional subscript c in $\mathcal{C}^k_c(\Omega, \mathbb{R}^n)$ we denote that the function has *compact support* in the domain Ω, which essentially means that the function values and all its derivates are zero at the domain boundary $\partial\Omega$ (see [13] for an exact definition).

3. Minimal Surface Reconstruction via Binary Segmentation

Everything should be made as simple as possible, but not simpler.

Albert Einstein
(1879-1955)

In this thesis we formulate image-based 3D reconstruction as a minimal surface problem. We are looking for a surface with minimal surface area which is subject to some constraints or certain boundary conditions. In our setup these constraints reflect information about the surface that has been extracted from one or several input images. The main idea and advantage behind this approach is to have a consistent way to deal with missing, redundant or even conflicting data that usually occurs in 3D reconstruction setups. Due to the constraints, the surface will align with the data as good as possible. In areas with redundant, but slightly different measurements we get an *interpolating* behavior of the surface, while in areas with missing data it will span a minimal surface and thus reflect an *extrapolating* behavior. Further, we will have the possibility to trade data alignment and surface smoothness with a single parameter in order to deal with noisy input data.

In this chapter we give a definition of minimal surfaces and show how they can be represented and efficiently computed by calculating the total variation of indicator functions.

3.1. Minimal Surfaces

Many equivalent definitions of minimal surfaces exist in the literature. We refer to Meeks and Perez [119] for an overview. In this work we make use of the following variational definition.

Definition 3.1 (Minimal Surface [119])**.** *A surface* $\Sigma \subset \mathbb{R}^n$ *is minimal if and only if it is a critical point of the area functional for all compactly supported variations.*

This means that any small variation of the minimal surface shape leads to an increase of the total surface area. In this sense minimal surfaces are a higher dimensional analogue to geodesics [41] which describe the shortest path between points on some embedded subspace. Figure 7.2 depicts the minimal surface of a soap bubble between two circles forming the shape of a catenoid.

In Part II (Single-View Reconstruction) of this thesis we look at surfaces being two-dimensional manifolds embedded into the three-dimensional space $\mathbb{R}^3$. Later, in Part III (Spatio-Temporal Multi-View Reconstruction),

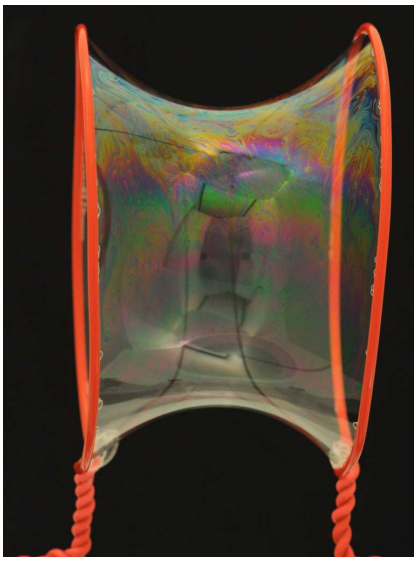


Figure 3.1.: Soap bubbles form minimal surfaces in order to connect the geometry they are applied to. In this example two parallel aligned rings are connected and form the shape of a catenoid. Image courtesy from [199].

we raise the dimensionality by including a temporal dependency and study three-dimensional manifolds embedded into the four-dimensional space $\mathbb{R}^4$.

3.2. Geometric Properties of the Total Variation

In this section we discuss important geometric properties of the total variation and how they can be used to compute non-parametric minimal surfaces based on perimeter minimization of sets that are represented via indicator functions.

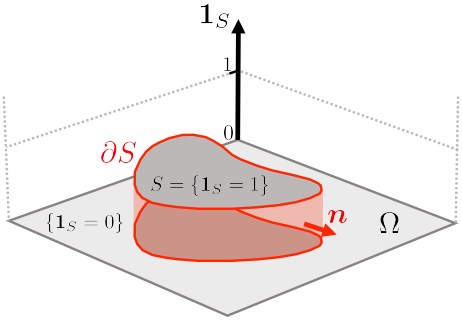


Figure 3.2.: Illustration of the indicator function $\mathbf{1}_S(\boldsymbol{x})$, the set $S \subset \Omega$, the set boundary ∂S and the normal $\boldsymbol{n}$ of the set boundary. The length of the sets' boundary curve $\mathcal{H}^{n-1}(\partial S)$ equals the perimeter of the set $\mathrm{Per}(S, \Omega)$ and the total variation of the sets' indicator function $\mathrm{TV}(\mathbf{1}_S, \Omega)$ because $\mathbf{1}_S$ jumps by 1 everywhere along the set boundary and nowhere else.

Definition 3.2 (Indicator function)**.** *Let $S \subseteq \Omega \subseteq \mathbb{R}^n$, then the indicator function $\mathbf{1}_S : \Omega \to \{0, 1\}$ of the set S is defined as*

$$\mathbf{1}_S(\boldsymbol{x}) := \begin{cases} 1 & \text{if } \boldsymbol{x} \in S \\ 0 & \text{if } \boldsymbol{x} \notin S \end{cases} \tag{3.1}$$

Definition 3.3 (Perimeter)**.** *Let $S \subset \Omega \subseteq \mathbb{R}^n$. The perimeter of the subset S in Ω is defined as*

$$\mathrm{Per}(S, \Omega) = \mathcal{H}^{n-1}(\partial S) = \mathrm{TV}(\mathbf{1}_S, \Omega) \ , \tag{3.2}$$

where $\mathcal{H}^{n-1}(\cdot)$ is the $(n-1)$-dimensional Haussdorff measure and ∂S is the boundary of the set S. This means that the perimeter of a set in dimension n is an $(n-1)$-dimensional measure of length, for instance, the length of the 1D-curve outlining a set in 2D space, or the 2D area of the surface in 3D space. The second equality holds because of the divergence theorem, which states that $-\int_S \mathrm{div}(\boldsymbol{p})\, d\boldsymbol{x} = \int_{\partial S} \boldsymbol{n} \cdot \boldsymbol{p}\, ds$. By Definition 2.12 of the TV we have for all vector fields $\boldsymbol{p} \in \mathcal{C}_c^1(\Omega, \mathbb{R}^n)$:

$$\begin{aligned}
\mathrm{TV}(\mathbf{1}_S, \Omega) &= \sup_{\|\boldsymbol{p}\|_\infty \leq 1} \Big\{ -\int_\Omega \mathbf{1}_S \cdot \mathrm{div}(\boldsymbol{p})\, d\boldsymbol{x} \Big\} && (3.3)\\
&= \sup_{\|\boldsymbol{p}\|_\infty \leq 1} \Big\{ -\int_S \mathrm{div}(\boldsymbol{p})\, d\boldsymbol{x} \Big\} && (3.4)\\
&= \sup_{\|\boldsymbol{p}\|_\infty \leq 1} \int_{\partial S} \boldsymbol{n} \cdot \boldsymbol{p}\, ds && (3.5)\\
&= \int_{\partial S} ds && (3.6)\\
&= \mathcal{H}^{n-1}(\partial S) \ , && (3.7)
\end{aligned}$$

since Equation (3.5) is maximized by any normalized vector field with $\boldsymbol{p}|_{\partial S} = \boldsymbol{n}$. The relationship of these measures for the 2D-case is depicted in Figure 3.2. Hence, we can easily compute this measure by evaluating the total variation of the sets' indicator function. Note that this result holds for any dimension n. As a result, minimizing the total variation of an indicator function $\mathbf{1}_\Sigma$ is equivalent to computing a minimal surface Σ.

The following two properties are important for optimization purposes and will later be used to show equivalence of TV-minimizers. One of them has been introduced by Fleming and

Rishel [82] in the following Theorem 3.4 which states that the total variation of a function equals the sum of the length of all its level lines.

Theorem 3.4 (Coarea formula [81, 82])**.** *Let function $u \in \mathcal{BV}(\Omega, \mathbb{R})$, then*

$$\mathrm{TV}(u, \Omega) = \int_{-\infty}^{+\infty} \mathrm{TV}(\mathbf{1}_{\{u \geq t\}}, \Omega)\, dt = \int_{-\infty}^{+\infty} \mathrm{Per}(\{u \geq t\}, \Omega)\, dt \ . \tag{3.8}$$

For a derivation and a proof of the theorem see [80, 13]. The following Theorem 3.5 is related to the previous one. It states that if the area under the function is sliced into horizontal layers (like a "layer-cake"), then the function value at any point $\boldsymbol{x}$ can also be computed as the sum of all these layers at $\boldsymbol{x}$. In mathematical terms this is called the sum of all level sets of function u at $\boldsymbol{x}$.

Theorem 3.5 (Layer-cake representation)**.** *Let u be a non-negative, real-valued, measurable function on Ω. Then*

$$u(\boldsymbol{x}) = \int_{0}^{\infty} \mathbf{1}_{\{u \geq t\}}(\boldsymbol{x})\, dt \ . \tag{3.9}$$

Proof. The formula can be transformed as follows

$$\int_{0}^{\infty} \mathbf{1}_{\{u(\boldsymbol{x}) \geq t\}}(\boldsymbol{x})\, dt = \int_{0}^{\infty} \mathbf{1}_{[0, u(\boldsymbol{x})]}(t)\, dt = \int_{0}^{u(\boldsymbol{x})} dt = \Big[\, t \,\Big]_{0}^{u(\boldsymbol{x})} = u(\boldsymbol{x}) \ . \tag{3.10}$$

□

3.3. Minimal Surfaces for 3D Reconstruction

Now we have all mathematical tools together to define the basis of our 3D reconstruction model. The key ingredient is the total variation, but in order to avoid trivial solutions further information in form of additional terms, constraints or boundary conditions is needed. In most cases, we will encode this information in form of a regional term model by means of a cost function f which locally favors either an interior or an exterior label. Then, the surface energy can be defined as follows.

Definition 3.6 (Non-parametric Minimal Surface)**.** *Let $\mathcal{M}(V)$ be the space of closed $(n-1)$-dimensional manifolds in V and $int(\Sigma)$ be the interior of surface Σ. Further, let function $f : V \to \mathbb{R}$ define the surface shape and $\lambda \in \mathbb{R}_{\geq 0}$ control its smoothness. Then a minimal surface is a minimizer of*

$$\Sigma^* \in \operatorname*{arg\,min}_{\mathcal{M}(V)} \left\{ \mathrm{Per}(\Sigma, V) + \lambda \int_{int(\Sigma)} f\, d\boldsymbol{x} \right\} \ . \tag{3.11}$$

Now we can make use of the total variation properties on indicator functions from the previous Section 3.2 and construct the following equivalent optimization problem. Let $u : V \subset \mathbb{R}^3 \to \{0, 1\}$, $u(\boldsymbol{x}) = \mathbf{1}_{int(\Sigma)}(\boldsymbol{x})$ be the binary labeling function indicating the interior or

exterior of the surface, then the following minimization problem is equivalent to the one in Equation (3.11):

$$u^* \in \underset{\mathcal{BV}(V,\{0,1\})}{\arg\min} \left\{ \mathrm{TV}(u, V) + \lambda \int_V f \cdot u \, d\boldsymbol{x} \right\} . \tag{3.12}$$

The indicator function defines a surface as the boundary of two disjunct subsets in $\mathbb{R}^3$, for instance a surface is the boundary between two different physical materials, such as air and wood or any other solid material.

Note that this implicit surface representation has several strong and desirable properties compared to other scene representations such as point clouds, mesh representations or other parametric surfaces like non-uniform rational B-splines (NURBS) [83]. The implicit surface representation can model arbitrary shapes of surfaces with an arbitrary topology (i.e. number of holes in an object) and an arbitrary number of surfaces (i.e. objects in the scene). Further, the implicit surface representation inherently defines a surface orientation, it disallows surface self-intersections and holes in the surface (i.e. surface boundaries), it makes the surface representation simple and independent of the object topology. A change of the object topology is implicitly handled, thus allowing a larger search space of feasible solutions with shapes of different topology. Moreover, it automatically assures a manifold with no boundaries which is often described as a 'watertight' surface in the literature.

Historical Note on Related Work on 3D Reconstruction via Minimal Surfaces. Minimal surfaces have already been studied quite early in 1760 by J.L.Lagrange [147] as the surface area of a two-dimensional function graph. Over the centuries many well-known mathematicians have contributed to this field (see [118] for more details).

Their application in computer vision began much later and has been influenced by adaptive object models for the purpose of object segmentation such as snakes, active contour models [128] and weighted geodesic contour models [132, 41] which describe the evolution of an object contour model that aligns with the image data. These approaches attracted much research and their generalization to the 3D domain was immediate [42, 43]. The major drawback of these models is the fixed topology, that is, the number of holes in the object either has to be fixed in advance or sophisticated splitting and merging techniques need to be applied.

This disadvantage has been tackled by level-set methods [169] which lift the problem to a space of higher dimension making topology changes natural and simple. Again, this approached had big impact on the computer vision community and was applied to 2D and 3D segmentation tasks, 3D reconstruction [79, 244] as well as spatio-temporal 3D reconstruction [95, 94].

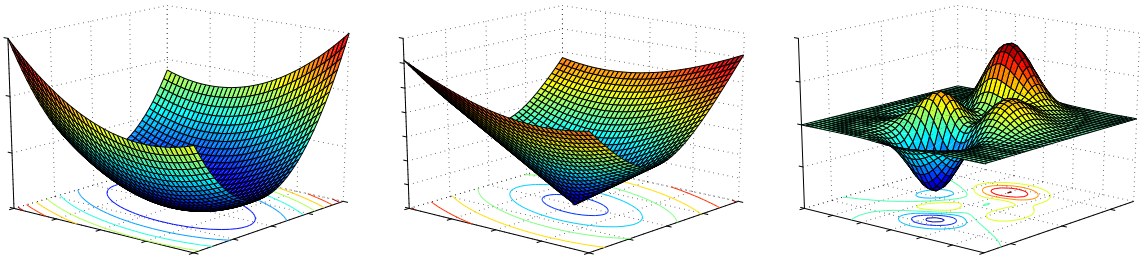
The main disadvantage of both, the active contour models and the level set approaches, is their strong dependence on proper initialization due to local optimization.

This has been changed by Chan, Esedoḡlu, and Nikolova [47] who proposed an implicit surface representation via indicator functions in the context of two-region image segmentation and most importantly showed the equivalence of minimizers for the binary and the corresponding relaxed optimization problem for their efficient computation. Later, this model has been extended to multi-view 3D reconstruction by Kolev et al. [135, 134] and formed the basis for follow-up works [216, 106].

4. Nonsmooth Convex Optimization

In fact the great watershed in optimization
isn't between linearity and nonlinearity,
but convexity and nonconvexity.

Ralph Tyrrell Rockafellar
(Mathematician and expert in optimization theory, 1935 - present)



smooth convex | nonsmooth convex | non-convex

Figure 4.1.: Example plots of a smooth convex, a nonsmooth convex and (smooth) non-convex function illustrating the difficulty of their global optimization in increasing order. The nonsmooth function is not differentiable at its minimizer. In contrast to the smooth convex function, gradient-based optimization methods are not directly applicable. Globally optimizing non-convex functions efficiently is an open research problem.

In this thesis we make constant use of convex optimization techniques and variational calculus. While convex optimization problems have important properties like duality theory and the property that any local minimum is also a global minimum, nonsmooth optimization deals with the problem of minimizing functions which are typically non-differentiable at their minimizers, see Figure 4.1 for an illustration. The total variation norm and its variants considered in this thesis share the properties of being convex and non-differentiable at zero which makes its minimization more difficult. However, powerful numerical algorithms have been developed in recent years and are briefly introduced in this chapter for their frequent use later in this thesis.

Respective articles will be cited along with the method's description, but there are also many good books providing an overview. The classic textbook on convex analysis is by Rockafellar [187]. The book of Boyd and Vandenberghe [26] gives an excellent introduction into convex optimization. Introductory books to variational calculus include [92, 65, 212, 188]. Books introducing variational calculus with focus and applications to image processing and computer vision are by Aubert and Kornprobst [14] and the German book by Bredies and Lorenz [33].

4.1. Convex Relaxation

In case a minimization problem is not directly solvable, a classical approach is to define a so called *relaxed* minimization problem that is substantially easier to solve and whose minima are equal or close to the minima of the original problem. While Aubert and Kornprobst [14]

provide a quite general definition of relaxation based on the convergence of sequences, in this thesis we only deal with one particular type of relaxation. Looking back at definitions in the previous Chapter 2, the *convex envelope* of the original non-convex functional is a convex functional that is closest to the original one and is thus the ideal candidate to pose a relaxed problem that is easier to solve. In general, however, it can be very difficult to find or compute the convex envelope of a non-convex functional.

In this thesis we mostly deal with convex functionals which are possibly constrained by convex constraints but defined on a non-convex domain. This is due to the binary inside-outside representation of the surface $u : V \to \{0, 1\}$ which is embedded in some 3-dimensional subspace $V \subset \mathbb{R}^3$. Hence we will deal with the following (hard) binary optimization problem.

$$u^*_{\text{bin}} \in \underset{u \in \mathcal{BV}(V,\{0,1\})}{\arg\min} \; E(u) \tag{4.1}$$

The binary function u^*_{bin} corresponds to a minimal surface being a critical point of some convex energy functional $E : \mathcal{BV}(V, \{0, 1\}) \to \mathbb{R}$ that assigns a real-valued cost to every BV-function. Relaxing the binary domain constraint by allowing function values on the full $[0, 1]$ interval makes the overall problem convex and thus much easier to solve.

$$u^*_{\text{rel}} \in \underset{u \in \mathcal{BV}(V,[0,1])}{\arg\min} \; E(u) \tag{4.2}$$

However, we now look at a different optimization problem and it is not clear how a minimizer of the relaxed problem u^*_{rel} relates to a minimizer of the binary problem u^*_{bin} that we actually want to compute. In the literature (e.g. [96]), one tries to find "tight" convex relaxations in a sense that they are close to the convex envelope of the original non-convex optimization problem. For some energies one can show that both, the non-convex and the corresponding relaxed problem are equivalent.

Equivalence of minimizers for certain functionals. Chan et al. [47] show for a widely usable problem class that the optimal solution of the binary problem can be obtained from the solution of the relaxed problem via simple pointwise thresholding. This is stated for the class of minimal surface energies in the following theorem.

Theorem 4.1 (Equivalence of Minimizers via Thresholding [204, 47])**.** *Let functional* $E : \mathcal{BV}(V, [0, 1]) \to \mathbb{R}$ *be of the form* $E(u) = \mathrm{TV}(u, V) + \lambda \int_V f u \, d\boldsymbol{x}$ *with function* $f : V \to \mathbb{R}$, $\lambda \in \mathbb{R}_{\geq 0}$ *and let*

$$u^*_{\text{rel}} \in \underset{u \in \mathcal{BV}(V,[0,1])}{\arg\min} \left\{ \mathrm{TV}(u, V) + \lambda \int_V f u \, d\boldsymbol{x} \right\} \tag{4.3}$$

be a global minimizer of the relaxed problem. Then, for any threshold value $\theta \in (0, 1)$ *the thresholded solution*

$$u_{\text{thr}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } u^*_{\text{rel}}(\boldsymbol{x}) \geq \theta \\ 0 & \text{if } u^*_{\text{rel}}(\boldsymbol{x}) < \theta \end{cases} \tag{4.4}$$

is a global minimizer of the corresponding binary minimization problem, that is,

$$u_{\text{thr}} = \mathbf{1}_{\{u^*_{\text{rel}} \geq \theta\}} \in \underset{u \in \mathcal{BV}(V,\{0,1\})}{\arg\min} \left\{ \mathrm{TV}(u, V) + \lambda \int_V f u \, d\boldsymbol{x} \right\} . \tag{4.5}$$

Proof. We proof the theorem by contradiction. Using the layer-cake representation in Equation (3.9) and the coarea formula in Equation (3.8) the energy in Equation (4.3) can be

expressed as

$$\begin{aligned} E(u) &= \mathrm{TV}(u, V) + \lambda \int_V f u \, d\boldsymbol{x} \\ &= \int_0^1 \left\{ \mathrm{TV}(\mathbf{1}_{\{u \geq \theta\}}, V) + \lambda \int_\Omega f \cdot \mathbf{1}_{\{u \geq \theta\}} \, d\boldsymbol{x} \right\} d\theta \\ &= \int_0^1 E(\mathbf{1}_{\{u \geq \theta\}}) \, d\theta \ . \end{aligned} \tag{4.6}$$

Since u^*_{rel} solves the relaxed problem, we have $E(u^*_{\mathrm{rel}}) \leq E(\mathbf{1}_{\{u^*_{\mathrm{rel}} \geq \theta\}})$. Now assume that the Theorem 4.1 does not hold, that is, there exists some set $\Sigma \subset V$ with a lower energy $E(\mathbf{1}_\Sigma) < E(\mathbf{1}_{\{u^*_{\mathrm{rel}} \geq \theta\}})$. We then derive

$$E(\mathbf{1}_\Sigma) = E(\mathbf{1}_\Sigma) \int_0^1 d\theta = \int_0^1 E(\mathbf{1}_\Sigma) \, d\theta < \int_0^1 E(\mathbf{1}_{\{u^*_{\mathrm{rel}} \geq \theta\}}) \, d\theta = E(u^*_{\mathrm{rel}}) \ , \tag{4.7}$$

which contradicts the fact that u^*_{rel} is a global minimizer of Equation (4.3). □

This result is not necessarily intuitive on the first sight as different thresholds will possibly lead to different binary solutions, but according to Theorem 4.1, possible different binary solutions will have the same energy and are all valid minimizers of the binary problem.

Energy bounds for the general case. For arbitrary functionals $E(u)$ the thresholding Theorem 4.1 does not hold in general. Nevertheless, it is useful to know how far any thresholded solution u_{thr} is away from the binary optimum u^*_{bin}. This can be computed based on the corresponding energies. Since $E(u^*_{\mathrm{bin}})$ is usually unknown, one can still give the following maximum energy bound on the energy distance between the thresholded and the optimal binary solution. The idea is also sketched in Figure 4.2.

Proposition 4.2 (Energy bounds for the distance to the optimum)**.** *Let u^*_{bin} be the global optimal solution of the binary problem, u^*_{rel} be the global optimal solution of the relaxed energy and $u_{\mathrm{thr}} = \mathbf{1}_{\{u^*_{\mathrm{rel}} \geq \theta\}}$ be a solution obtained by thresholding u^*_{rel} at θ. Then the following relation holds:*

$$E(u_{\mathrm{thr}}) - E(u^*_{\mathrm{bin}}) \leq E(u_{\mathrm{thr}}) - E(u^*_{\mathrm{rel}}) \ . \tag{4.8}$$

Proof. Equation (4.8) directly follows from the fact $E(u^*_{\mathrm{bin}}) \geq E(u^*_{\mathrm{rel}})$. □

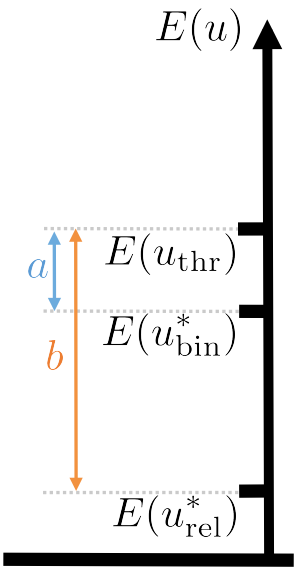


Figure 4.2.: Illustration of the energy bound. Since u^*_{bin} is unknown, the energy difference a is unknown too, but it is bounded by the energy difference $b \geq a$. So b is a worst case estimate of a.

4.2. Properties of Optimization Problems

4.2.1. Constrainted Optimization Problems

We now discuss methods for dealing with optimization problems that are subject to a set of constraints. The definitions and notations mostly follow the ones in the book of Boyd and Vandenberghe [26, page 215ff] which we also recommend for further reading.

Let function $u \in \mathcal{BV}(V, \mathbb{R})$ and let $\{E_i : \mathcal{BV}(V, \mathbb{R}) \to \mathbb{R}\}_{i=0}^{m}$ and $\{H_i : \mathcal{BV}(V, \mathbb{R}) \to \mathbb{R}\}_{i=1}^{p}$ be sets of functionals of which E_0 is the objective and the remaining functionals form sets of m inequality constraints $\{E_i(\boldsymbol{x}) \leq 0\}_{i=1}^{m}$ and p equality constraints $\{H_i(\boldsymbol{x}) = 0\}_{i=1}^{p}$. Consider the following constrained optimization problem in normal form

$$\begin{aligned} u^* = \underset{u \in \mathcal{BV}(V,\mathbb{R})}{\arg\min} \;\; & E_0(u) \\ \text{s.t. } & E_i(u) \leq 0, \qquad i = 1, \dots, m \\ & H_i(u) = 0, \qquad i = 1, \dots, p \;\; . \end{aligned} \tag{4.9}$$

This optimization problem will be called the *primal* problem and might also be written as follows

$$u^* = \underset{u \in U_C}{\arg\min} \; E_0(u) \tag{4.10}$$

$$\text{with} \quad U_C = \Big\{ u \in \mathcal{BV}(V, \mathbb{R}) \;\Big|\; E_i(u) \leq 0 \;\; \forall i = 1, \dots, m \,, \;\; H_i(u) = 0 \;\; \forall i = 1, \dots, p \Big\} \;\; .$$

The set U_C is called the *feasible set* of the optimization variable u. This notation already gives an intuition about one possible way to deal with the constraints: We can numerically minimize E_0, for instance with gradient descent, and project onto the set U_C after each iteration, leading to a projected gradient descent. These numerical optimization methods are later explained in Section 4.3. Essentially, one alternates between ignoring and enforcing the constraints during the numerical optimization. However, this approach requires the projection onto the set U_C to be feasible and practicable.

Another possible way is to transform the *constrained* optimization problem (4.10) into an *unconstrained* optimization problem by augmenting the objective function with a weighted sum of the constraint functions

$$L(u, \boldsymbol{\lambda}, \boldsymbol{\nu}) = E_0(u) + \sum_{i=1}^{m} \lambda_i E_i(u) + \sum_{i=1}^{p} \nu_i H_i(u) \;\; , \tag{4.11}$$

where function $L : \mathcal{BV}(V, \mathbb{R}) \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is called the *Lagrangian* and the new variables $\boldsymbol{\lambda} = (\lambda_1 \cdots \lambda_m)^T$, $\boldsymbol{\nu} = (\nu_1 \cdots \nu_p)^T$ - exactly one for each constraint - are called *Lagrangian multipliers*.

Lagrangian dual function. By minimizing the Lagrangian in Equation (4.11) over the *primal* variable u yields a new functional that only depends on *dual* variables $\boldsymbol{\lambda}, \boldsymbol{\nu}$:

$$G(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{u \in \mathcal{BV}(V,\mathbb{R})} L(u, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{u \in \mathcal{BV}(V,\mathbb{R})} \left\{ E_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i E_i(\boldsymbol{x}) + \sum_{i=1}^{p} \nu_i H_i(\boldsymbol{x}) \right\} \;\; , \tag{4.12}$$

The dual function provides lower bounds on the optimal value u^* of the primal problem, that is, for any $\boldsymbol{\lambda}$ and any $\boldsymbol{\nu} \succeq 0$ it holds that $G(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq u^*$. More importantly for our purposes,

if the primal problem is convex, then this bound is tight and we have

$$\sup_{\boldsymbol{\lambda},\boldsymbol{\nu}\succeq 0} G(\boldsymbol{\lambda},\boldsymbol{\nu}) = \sup_{\boldsymbol{\lambda},\boldsymbol{\nu}\succeq 0} \inf_{u\in\mathcal{BV}(V,\mathbb{R})} L(u,\boldsymbol{\lambda},\boldsymbol{\nu}) = \inf_{u\in U_C} E_0(u) \ , \tag{4.13}$$

Soft constraints. Related to the method of Lagrangian multipliers is the idea of adding the equality constraints with a scalar weight that controls how much the corresponding constraint should be enforced and provides an order about the priority among all constraints:

$$E(u;\boldsymbol{\nu}) = E_0(u) + \sum_{i=1}^{p} \nu_i \left|H_i(u)\right|^d \quad \text{for fixed } \boldsymbol{\nu} \text{ and } d\in\{1,2\} \ , \tag{4.14}$$

where the absolute value and parameter d ensures that possible negative values of $H_i(u)$ translate into positive costs when minimizing the overall energy. For infinitely large weights $\nu_1,\cdots,\nu_p$ the constraint terms govern the minimization as any constraint violation immediately yields a suboptimal result. Conversely, for smaller weights a constraint violation has less impact on the overall energy and the constraints are enforced in a "soft" manner, in a sense that constraint-compliant solutions are preferred but not enforced.

Duality theory provides important results on how optima of *primal* and *dual* problems are characterized, how they are related and how they can be computed. However, as we have seen in Equation (4.13), instead of minimizing the original *primal* problem, or maximizing the *dual* problem one can also solve a *saddle point* problem which depends on both the primal and the dual variables.

4.2.2. Saddle Point Problems

For optimization purposes (described in the next Section 4.3), we are mainly interested in computing saddle points of *convex-concave* functionals.

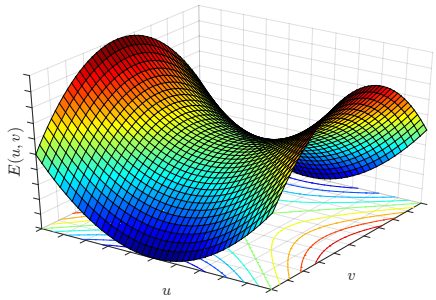


Figure 4.3.: A saddle point problem. The functional $E(u,v)$ is convex in variable u and concave in variable v.

Definition 4.3 (Convex-Concave Functional)**.** *A functional $E : U\times\mathcal{V}\to\mathbb{R}\cup\{-\infty,\infty\}$ on the vector spaces U and $\mathcal{V}$ is called convex-concave if*

- *the mapping $\forall v\in\mathcal{V}: \ u\mapsto E(u,v)$ is either convex or constant $-\infty$, and*
- *the mapping $\forall u\in U: \ v\mapsto E(u,v)$ is either concave or constant ∞.*

Definition 4.4 (Saddle Point)**.** *Let $E : U\times\mathcal{V}\to\mathbb{R}\cup\{\infty\}$ be a functional on the vector spaces U and $\mathcal{V}$. The tuple $(u^*,v^*)\in U\times\mathcal{V}$ is called a saddle point of E if*

$$u^*\in\arg\min_{u\in U} E(u,v^*) \quad \text{and} \quad v^*\in\arg\max_{v\in\mathcal{V}} E(u^*,v) \tag{4.15}$$

As we have just seen, saddle point problems arise by transforming constrained optimization problems or computing convex conjugates. For the optimization problems we consider in this thesis, optimization algorithms which directly operate on saddle point energies are currently among the fastest ones. More information on duality theory and saddle points can be found in [187, 26].

4.2.3. Extremality Conditions

As for analyzing and finding extrema of functions similar properties also hold for functionals. An important property is the necessary condition for a minimum value of a differentiable functional given in the following definition.

Definition 4.5 (Euler-Lagrange equation)**.** *Considering continuously differentiable functionals of the form* $E(u) = \int_V \mathcal{L}(\boldsymbol{x}, u, \nabla u)\, d\boldsymbol{x}$*, with* $\mathcal{L} : V \times \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ *and domain* $V \subseteq \mathbb{R}^n$*. Then, the necessary condition for a minimum of the functional is given by the corresponding Euler-Lagrange equation*

$$0 \quad = \quad \frac{dE}{du} \quad = \quad \frac{\partial \mathcal{L}}{\partial u} - \sum_{i=1}^{n} \frac{d}{d\boldsymbol{x}_i} \frac{\partial \mathcal{L}}{\partial u_{\boldsymbol{x}_i}} \ , \tag{4.16}$$

in which $\mathcal{L}(\cdot)$ *is called the Lagrangian density. Further, we used the shorthand notation* $u_{\boldsymbol{x}_i} = \frac{\partial u}{\partial \boldsymbol{x}_i}$*.*

Equation (4.16) usually describes a partial differential equation (PDE) [78] that needs to be fulfilled. Solving this equation with respect to u is also a common way to compute minimizers of the energy.

Saddle-Point Problems. Similarly, the first order condition that the local gradient vanishes must also hold for saddle point problems, but for both variables.

Proposition 4.6. *Let function* $E : U \times \mathcal{V} \to \mathbb{R} \cup \{-\infty, \infty\}$ *be convex-concave, then the point* (u^*, v^*) *is a saddle point if and only if*

$$0 \in \partial_u E(u^*, v^*) \quad \text{and} \quad 0 \in \partial_v E(u^*, v^*) \ , \tag{4.17}$$

where ∂_x *denotes the subgradient of* E *with respect to* x*.*

4.3. Algorithms for Total Variation Minimization

In this section we discuss several methods for minimizing the total variation norm in conjunction with convex data terms and possible additional constraints. This will be the basis for the numerical computation of solutions to all the optimization problems posed in this thesis.

The first two of the following minimization algorithms, namely "gradient descent" and the "lagged diffusivity fixed point iteration" approach by [231], require differentiability of the objective function. In order to deal with the non-differentiability of the TV-norm Rudin, Osher, Fatemi [191] suggested a slight perturbation of the TV-norm:

$$\mathrm{TV}(u, \Omega)_\epsilon = \int_\Omega |\nabla u|_\epsilon \, d\boldsymbol{x} \qquad \text{with} \quad |\nabla u|_\epsilon = \sqrt{|\nabla u|_2^2 + \epsilon} \ , \tag{4.18}$$

where $\epsilon > 0$ is a small positive number. The choice of this number is a trade-off between the numerical stability of the method and the accuracy of the result. Note that the other optimization methods considered in this section do not need such an approximation as they directly deal with the non-differentiability. Further note, that there exist a lot more methods and variants for minimizing TV-based functionals, e.g. Split-Bregman methods [98], or the alternating direction method of multipliers (ADMM) [76, 159] which are not detailed in this thesis.

4.3.1. Gradient Descent

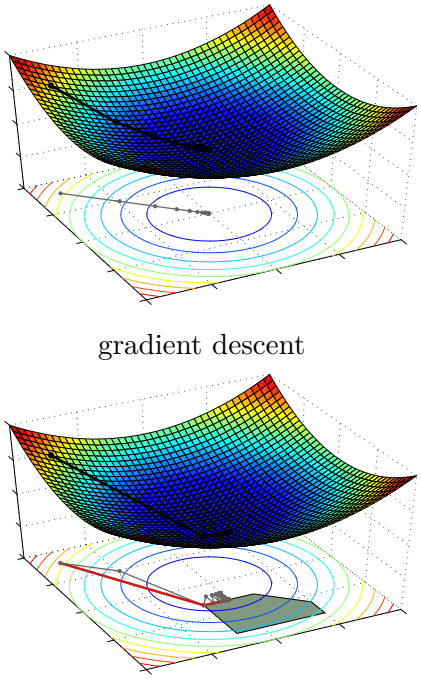
Gradient descent is one of the most basic minimization algorithms which can only be applied if the gradient of the function to be minimized can be computed, that is, the energy $E(u)$ needs to be differentiable in u. Since the gradient of the energy function dE/du locally indicates in which way u has to be changed in order to lower the energy $E(u)$, the main idea is to use this property within an evolutionary process over time.

$$\frac{du}{dt} = -\frac{dE}{du} \tag{4.19}$$

If we interpret the energy E as a mountain range over the space of u, this equation states that for every time change, the energy should decrease by the negative gradient $-dE/du$ which points towards the steepest descent direction. Hence, this process corresponds to a downhill walk. The approximation of the temporal derivative by a forward difference $du/dt \approx (u^{k+1} - u^k)/\tau$ yields an iterative update scheme. Starting with an initial solution u^0 the algorithm iterates

$$u^{k+1} = u^k - \tau \frac{dE(u^k)}{du} \tag{4.20}$$

and converges to the next local minimum or saddle. Possible convergence criteria are discussed later in Section 4.3.5. The choice of the step size $\tau > 0$ steers both speed and stability of the method. Large step sizes easily lead to oscillation or unstable behavior while small step sizes make the method extremely slow. The best trade-off between these contrary goals depends on the particular problem instance and is not easy to find. Compared to other algorithms, if applicable, gradient descent is usually among the slowest methods. An example plot of gradient descent iterations is shown in Figure 4.4.



gradient descent

projected gradient descent

Figure 4.4.: Gradient descent steps on a quadratic function with and without constraints on the feasible domain (light gray). The bottom figure shows algorithm iterates after each combined descent and projection step (black and red). Separate steps for the descent step and the projection are plotted in dark gray below the graph.

Projected Gradient Descent is a generalization of the gradient descent method to minimize functions on a restricted domain U_C (also called feasible domain). To this end, the update step in Equation (4.20) is augmented with a projection step onto the feasible domain. The method is summarized in Algorithm 1. For unrestricted domains, the projection is the identity operation $\Pi_{U_C}(u) = u$ and the method reduces to the ordinary gradient descent. Figure 4.4

Algorithm 1 Projected Gradient Descent (PGD)

Input : initial value $u^0 \in U$
Output: locally optimal value $u^* \in U_C$
loop until convergence

$$u^{k+1} = \Pi_{U_C}\left[u^k - \tau \frac{dE(u^k)}{du}\right] \tag{4.21}$$

end loop

illustrates gradient descent steps in comparison with the ones from projected gradient descent. The bottom figure shows the case that the function minimum is not in the feasible area, then the gradient step leads outside the feasible set and the projection step yields again a feasible point at the boundary of the feasible set.

4.3.2. Lagged Diffusivity Fixed Point Iterations

In [231], Vogel and Oman proposed the lagged diffusivity fixed point iteration (LDFPI) scheme for minimizing a total variation-based cost function. For our purposes we consider TV together with a linear data term $f : V \to \mathbb{R}$, but in principle this algorithm works with any additional term that has a linear derivative.

$$u^* = \arg\min_{u \in U} \left\{ \int_V |\nabla u| \, d\boldsymbol{x} + \lambda \int_V f u \, d\boldsymbol{x} \right\} \tag{4.22}$$

The extremality condition is given by zeroing the corresponding Euler-Lagrange equation

$$\lambda f - \operatorname{div}(g \nabla u) = 0 \qquad \text{with} \quad g = \frac{1}{|\nabla u|_\epsilon} \tag{4.23}$$

which corresponds to a diffusion equation and function g, called diffusivity, steers the amount of diffusion in every point. The perturbed norm $|\cdot|_\epsilon$ avoids a division by zero and is defined above in Equation (4.18). The diffusivity g is the only source of nonlinearity in Equation (4.23) for the case that g locally depends on u. The key idea is to neglect this dependency for a moment and treat g as a constant. Then Equation (4.23) describes a linear system of equations which can be solved efficiently. In order to deal with the nonlinearity of g, the problem of solving the linear system with constant g is embedded within an outer fixed point iteration in which g is recomputed based on the current estimate of u. In this sense, the value of the diffusivity g is lagging behind, because for solving the linear system the previous value of g is always used. Chan et al. [49] proved the linear convergence of this algorithm.

Hence, a linear system needs to be solved for every fixed point step. Vogel-Oman [232] use a preconditioned conjugate gradient solver for that purpose. Similar to Kolev et al. [135] we use Successive Over-Relaxation (SOR) [245] because of its quick convergence and its suitability for parallelization. With the SOR update equations, the outer fixed point iteration scheme can be shown to correspond to a Quasi-Newton method (see [134, Proposition 2]). After

Algorithm 2 Lagged Diffusivity Fixed Point Iterations (LDFPI)

Input : initial value $u^0 \in U$
Output: globally optimal value $u^* \in U$

loop until convergence ▷ fixed-point iteration
  $g^k \leftarrow g(u^k)$ ▷ re-compute diffusivities
  $u^{k+1} \leftarrow$ solve: $0 = \lambda f - \operatorname{div}(g^k \nabla u^{k+1})$ ▷ solve linear system with fixed diffusivities
end loop

discretizing function u and the differential operators the Euler-Lagrange equation with fixed diffusivities in Algorithm 2 can be written as the linear system $0 = Au - b$ in which A is a sparse matrix containing the discretized differential operators. The sparsity of A makes the implementation and parallelization of the SOR algorithm easy, as been described in the following.

Gauss-Seidel and Successive Over-Relaxation Algorithms aim to solve a linear system of equations

$$A\boldsymbol{x} = b \tag{4.24}$$

with respect to vector $\boldsymbol{x} \in \mathbb{R}^n$ for a given vector $b \in \mathbb{R}^n$ and a given matrix $A \in \mathbb{R}^{n\times n}$ with $A = L_* + U$ being split into a lower triangular matrix with diagonal entries L_* and the upper triangular matrix U.

$$L_* = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ a_{n1} & \dots & a_{n,n-1} & a_{nn} \end{pmatrix} \tag{4.25}$$

$$U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \dots & 0 & 0 \end{pmatrix} \tag{4.26}$$

By using the decomposition of matrix A and by introducing a temporal dependency of vector $\boldsymbol{x}$, Equation (4.24) can be cast into an iterative numerical update scheme.

$$(L_* + U)\boldsymbol{x} = b \tag{4.27}$$
$$\boldsymbol{x} = L_*^{-1}(b - U\boldsymbol{x}) \tag{4.28}$$
$$\boldsymbol{x}^{k+1} = L_*^{-1}(b - U\boldsymbol{x}^k) \tag{4.29}$$

This is the Gauss-Seidel algorithm which is guaranteed to converge as long as matrix A is either diagonally dominant, that is $\forall i : |a_{ii}| \geq \sum_{j\neq i} |a_{ij}|$, or symmetric and positive definite [17].

The algorithm can be further accelerated by the following simple linear extrapolation step.

$$\boldsymbol{x}^{k+1} = (1-\omega)\boldsymbol{x}^k + \omega\bar{\boldsymbol{x}}^{k+1} \tag{4.30}$$

For any interpolation or extrapolation variable $\omega \in (0, 2)$ the algorithm is proven to converge [124, 36]. In our experiments, values $\omega \in [1.5, 1.9]$ gave the best performance.

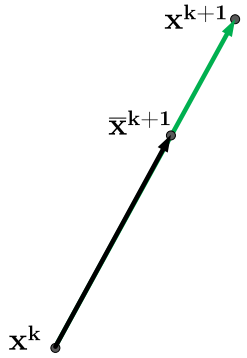


Figure 4.5.: Linear extrapolation in Successive Over-relaxation: The algorithm step from x^k to $\bar{x}^{k+1}$ is further extrapolated to x^{k+1}.

Algorithm 3 Successive Over-Relaxation (SOR)

Input : initial value $\boldsymbol{x}^0 \in \mathbb{R}^n$
Output: solution $\boldsymbol{x}^* \in \mathbb{R}^n$

loop until convergence

$$\bar{\boldsymbol{x}}_i^{k+1} = \frac{1}{a_{ii}}\Big(b_i - \underbrace{\sum_{j>i} a_{ij}\boldsymbol{x}_j^k}_{U} - \underbrace{\sum_{j<i} a_{ij}\boldsymbol{x}_j^{k+1}}_{L_*}\Big) \qquad \forall i \in \{1, \dots, n\} \tag{4.31}$$

$$\boldsymbol{x}^{k+1} = (1-\omega)\boldsymbol{x}^k + \omega\bar{\boldsymbol{x}}^{k+1} \tag{4.32}$$

end loop

4.3.3. Fast Iterative Shrinkage and Thresholding Algorithm

Many problems in computer vision consist of minimizing the sum of two terms, mostly a data term and a regularization term. In [20], Beck and Teboulle proposed an efficient algorithm

for minimizing the sum of two convex functions

$$\min_{u \in U} F(u) + G(u) \tag{4.33}$$

for a smooth function $G(u)$ and a possibly nonsmooth function $F(u)$. The algorithm is an accelerated variant based on a class of iterative shrinkage and thresholding algorithms (see for example [66]) which have a runtime complexity of $\mathcal{O}(1/k)$. In contrast, the accelerated FISTA is proven to converge in $\mathcal{O}(1/k^2)$ time. The first step in Equation (4.34) of the algorithm

Algorithm 4 Fast Iterative Shrinkage and Thresholding Algorithm (FISTA)

Input : initial value $u_0 \in U$, upper bound $L \geq L(G)$ on the Lipschitz constant $L(G)$ of ∇G.

Output: globally optimal value $u^* \in U$

initialize $\bar{u}^1 = u^0$, $\tau^k = 1$

loop until convergence

$$u^k = \operatorname{prox}_{L^{-1}F}\left(\bar{u}^k - \frac{1}{L}\nabla G(\bar{u}^k)\right) \tag{4.34}$$

$$\tau^{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4(\tau^k)^2}\right) \tag{4.35}$$

$$\bar{u}^{k+1} = u^k + \left(\frac{\tau^k - 1}{\tau^{k+1}}\right)(u^k - u^{k-1}) \tag{4.36}$$

end loop

is a gradient descent step in the differentiable component G and a subsequent subgradient descent step in the non-differentiable component F. Step three in Equation (4.36) is a linear extrapolation step similar to the one in the Successive Over-Relaxation Algorithm 3 and the step before Equation (4.35) computes the step width of the extrapolation adaptively. The proximity operator $\operatorname{prox}_{\tau G}(\cdot)$ (e.g. see [58]) implicitly performs a subgradient descent step of step size τ on the functional G and is defined as

$$\operatorname{prox}_{\tau G}(u) := \arg\min_{v} \left\{ \frac{1}{2}\|u - v\|^2 + \tau G(v) \right\} \quad . \tag{4.37}$$

4.3.4. First Order Primal-Dual Algorithm

In [173, 46], Chambolle and Pock suggested an algorithm for minimizing the sum of two convex functions one of which is allowed to be non-differentiable.

$$\min_{u \in U} F(Ku) + G(u) \tag{4.38}$$

Both functions $F, G : U \to \mathbb{R}$ need to be proper, convex, lower-semicontinuous functions. The input of the function F is transformed by a linear operator $K : U \to P$ and F itself can be non-differentiable. In most of our applications, F will represent the non-differentiable total variation regularizer by the 2-norm and $K = \nabla$ being the gradient operator. The corresponding adjoint operator K^* (Definition 2.8) can be found via Definition 2.11 to be $K^* = -\operatorname{div}$.

Using the definitions of the convex conjugate (Definition 2.6) and the adjoint operator (Definition 2.8) the so called *primal* minimization problem in Equation (4.38) can be transformed into the following equivalent *(*primal-dual) saddle point problems (see Section 4.2.2) or the

dual maximization problem:

$$\min_{u \in U} F(Ku) + G(u) \qquad \text{(primal)} \tag{4.39}$$

$$= \min_{u \in U} \max_{\boldsymbol{p} \in P} \langle Ku, \boldsymbol{p} \rangle - F^*(\boldsymbol{p}) + G(u) \qquad \text{(primal-dual)} \tag{4.40}$$

$$= \max_{\boldsymbol{p} \in P} \min_{u \in U} \langle u, K^*\boldsymbol{p} \rangle - F^*(\boldsymbol{p}) + G(u) \qquad \text{(dual-primal)} \tag{4.41}$$

$$= \max_{\boldsymbol{p} \in P} -(F^*(\boldsymbol{p}) + G^*(-K^*\boldsymbol{p})) \qquad \text{(dual)} \tag{4.42}$$

The algorithm in [173, 46] solves the *primal-dual* saddle point problem in Equation (4.40) by iterating the numerical update scheme in Algorithm 5. The first two update steps represent

Algorithm 5 First Order Primal-Dual (PD) Algorithm

Input : initial values $u^0 \in U, \boldsymbol{p}^0 = 0$
Output: globally optimal value $u^* \in U$

loop until convergence

$$\begin{aligned} \boldsymbol{p}^{k+1} &= \text{prox}_{\sigma F^*}\left(\boldsymbol{p}^k + \sigma K \bar{u}^k\right) \\ u^{k+1} &= \text{prox}_{\tau G}\left(u^k - \tau K^* \boldsymbol{p}^{k+1}\right) \\ \bar{u}^{k+1} &= u^{k+1} + \theta(u^{k+1} - u^k) \end{aligned} \tag{4.43}$$

end loop

a projected gradient descent in the primal variable and the projected gradient ascent in the dual variable. The third step is an extrapolation step which is needed to guarantee algorithm convergence. Moreover, the step sizes $\sigma > 0, \tau > 0$ need to be sufficiently small, in particular $\tau\sigma L^2 < 1$ with L being the Lipschitz constant of F which can be estimated by the Frobenius norm of operator K, i.e. $L = \|K\|$.

The algorithm also makes use of the proximity operator (defined in Equation (4.37)) which is a generalization of a projection operator onto a convex set. If the function G of the proximity operator $\text{prox}_G(x)$ is the characteristic function of a convex set C, that is, $G(x) = \chi_C(x)$, where $\chi_C(x) := \{0 \text{ if } x \in C, \infty \text{ else}\}$, then the proximity operator simplifies to the Euclidean projection, denoted as Π_C, onto the set C. Thus, $\text{prox}_{\chi_C}(x) = \Pi_C(x)$ (please refer to [58] for more details).

Primal-Dual Gap. The *dual* maximization problem in Equation (4.42) is useful to analyze the algorithms' convergence, because the algorithm does not necessarily decrease the primal energy or increase the dual energy in every step. Instead the algorithm minimizes the difference of both energies, called the primal-dual gap, defined as

$$\text{Gap}(u, \boldsymbol{p}) = F(Ku) + G(u) + F^*(\boldsymbol{p}) + G^*(-K^*\boldsymbol{p}) \ . \tag{4.44}$$

Preconditioning. In a follow-up work, Pock and Chambolle [175] suggested a better way for choosing the step sizes in the update steps in order to improve the convergence speed of the algorithm. In a slightly modified update scheme, the real-valued step sizes τ, σ are replaced by corresponding matrices T, Σ which can provide different step sizes for each dimension. In particular, Lemma 2 in [175] suggests to use diagonal preconditioners by choosing the primal and dual step sizes as follows. Let K be an $m \times n$ matrix with $n = \dim U$ and $m = \dim P$,

then choose $T = \text{diag}(\tau_1, \ldots, \tau_n)$ and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_m)$ with

$$\tau_j = \frac{1}{\sum_{i=1}^{m} |K_{i,j}|^{2-\alpha}} \quad , \qquad \sigma_i = \frac{1}{\sum_{j=1}^{n} |K_{i,j}|^{\alpha}} \tag{4.45}$$

for any $\alpha \in [0, 2]$. Throughout this thesis we always used $\alpha = 1$. In sum, the preconditioned primal-dual algorithm is a slight modification of Algorithm 5. The necessary conditions for

Algorithm 6 Preconditioned First Order Primal-Dual (PD) Algorithm

Input : initial values $u^0 \in U, \boldsymbol{p}^0 = 0$
Output: globally optimal value $u^* \in U$

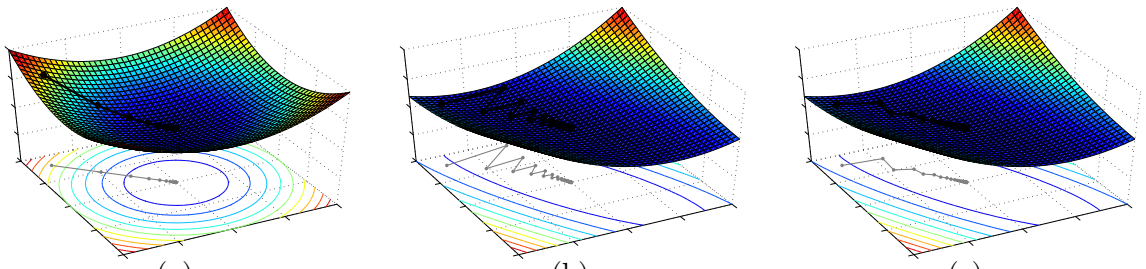
loop until convergence

$$\begin{aligned} \boldsymbol{p}^{k+1} &= \text{prox}_{\Sigma F^*}\left(\boldsymbol{p}^k + \Sigma K \bar{u}^k\right) \\ u^{k+1} &= \text{prox}_{TG}\left(u^k - TK^* \boldsymbol{p}^{k+1}\right) \\ \bar{u}^{k+1} &= u^{k+1} + \theta(u^{k+1} - u^k) \end{aligned} \tag{4.46}$$

end loop

the algorithms' convergence are still valid and faster convergence rates have been empirically shown for several computer vision applications [175].

Essentially, the step size for each dimension is scaled by the operator norm of the corresponding matrix row $K_{i,*}$ or matrix column $K_{*,j}$, respectively. This choice of step size acts like a normalization and equalizes the step sizes among different dimensions with respect to the function scale. The intuition behind this scaling is the fact that a gradient descent converges very fast if the shape of the cost function is isotropic, because the gradient direction always points towards the global optimum as shown in Figure 4.6 (a). If the cost function has an anisotropic shape the convergence is slower because the gradient direction does not necessarily point towards the minimum. Depending on the cost function and the chosen step size this may lead to the typical zig-zag pattern of gradient descent iterations, see Figure 4.6 (b). For the optimization problem in Equation (4.38), matrix K has large influence on the anisotropy of the overall function shape. With Equation (4.45) this information is used to rescale the step sizes independently for each dimension to get a convergence behavior that is closer to the isotropic case and usually leads to faster convergence as illustrated in Figure 4.6 (c).



(a) (b) (c)

Figure 4.6.: Plots of gradient descent iterations on an isotropic and an anisotropic convex function schematically illustrate the principle of the preconditioning [175] for the primal-dual algorithm. (a) gradient descent quickly approaches the minimum on convex functions with isotropic shape, (b) the same convex function has been scaled in one dimension leading to slower convergence due to zig-zag patterns in the gradient descent, (c) proper dimension-wise scaling of the step sizes improves the convergence rate.

Many optimization problems in computer vision can be cast into the form of Equation (4.38). For this problem class the preconditioned primal dual algorithm is currently among the fastest ones available for efficient minimization, especially if each update step in Equation (4.46) can be parallelized over the function domain. On the downside, this algorithm usually needs more memory due to the additional dual variable and extra storage for a copy of the primal variable needed for the extrapolation step.

4.3.5. Convergence Criteria

For the optimization algorithms discussed above different convergence criteria are possible and found in the literature. In this section we briefly discuss their properties. The following convergence criteria have been considered in this thesis:

1. **Fixed number of iterations.** The number of iteration depends on the optimization algorithm and the particular problem instance and should thus be chosen to be sufficiently large. It is usually necessary to also check one of the following criteria.
2. **Energy based.** For methods which reduce the energy in every iteration, such as gradient descent, it is reasonable to check if the percental energy change between consecutive iterations falls below a predefined threshold θ_E.

$$\left| \frac{E(u^{k-1}) - E(u^k)}{E(u^k)} \right| < \theta_E \tag{4.47}$$

   This scheme is not suited for the primal-dual algorithm because it minimizes the difference between primal and dual energies, that is, the primal dual gap (Equation (4.44)). Each of these energies may decrease or increase in arbitrarily small amounts between two iterations. Therefore, for the primal-dual algorithm the scheme above should be modified to check the convergence of the primal-dual gap rather than the primal energy:

$$\left| \frac{\mathrm{Gap}(u^{k-1}, \boldsymbol{p}^{k-1}) - \mathrm{Gap}(u^k, \boldsymbol{p}^k)}{\mathrm{Gap}(u^k, \boldsymbol{p}^k)} \right| < \theta_{\mathrm{Gap}} \ . \tag{4.48}$$

3. **Solution based.** A simple convergence check is look at the percental change between two consecutive solutions:

$$\frac{|u^k - u^{k+1}|}{|u^k|} < \theta_U \ . \tag{4.49}$$

   Even for large problems, this criterion is usually very cheap to compute because the current solution u^k is computed anyway. Note that this method is not robust to oscillations, which may for instance occur for large step sizes in the gradient descent scheme. This can be tackled by combining the scheme with one of the above ones.

The energy computations in Equations (4.47) and (4.48) are usually costly to compute. To keep the numerical optimization fast, a common remedy is to the check convergence only after a set of iterations of the algorithm rather than in every iteration.

4.4. Discretization

Since we formulate our approaches in a continuous setting we still need to discretize all equations and operators for their numerical implementation. A legitimate question is why we describe all theory in a continuous setting and not directly on a discrete grid. A continuous formulation has several advantages: 1) the theory is independent of the choice of the

underlying grid and only the discretization needs to be changed when switching to another grid representation. 2) Some continuous formulations are simpler and more general (e.g. rotational invariance with respect to the input data). 3) Discrete approaches suffer from metrication errors stemming from the underlying grid structure which do not vanish in the limit if the grid size is refined (see Klodt et al. [133]).

For the discretization of images, we consider a two dimensional regular Cartesian grid of size $N \times M$ with equidistant grid spacing $h_\Omega \in \mathbb{R}_{\geq 0}$

$$\Omega = \left\{ (i \cdot h_\Omega, j \cdot h_\Omega) \in \mathbb{R}^2 \;\middle|\; i, j \in \mathbb{Z},\ 1 \leq i \leq N,\ 1 \leq j \leq M \right\} . \tag{4.50}$$

Likewise, we use a three dimensional regular Cartesian grid of size $O \times P \times Q$ for the discretization of the volume with grid spacing $h_V \in \mathbb{R}_{\geq 0}$:

$$V = \left\{ (i \cdot h_V, j \cdot h_V, k \cdot h_V) \in \mathbb{R}^3 \;\middle|\; i, j, k \in \mathbb{Z},\ 1 \leq i \leq Q,\ 1 \leq j \leq P,\ 1 \leq k \leq Q \right\} . \tag{4.51}$$

In this thesis we use simple finite difference schemes to approximate differential operators on a discrete grid, because for our setting they offer the best trade-off between computational efficiency and sufficient approximation accuracy (see [45] for more information).

In the following we describe the discretization of the differential operators for the three-dimensional case. The two- and four-dimensional cases are analog. The gradient operator $\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}\right)^T$ is discretized as

$$(\nabla u)_{i,j,k} = \left(\delta_x^+ u_{i,j,k}, \delta_y^+ u_{i,j,k}, \delta_z^+ u_{i,j,k}\right)^T \tag{4.52}$$

discrete derivatives for the gradient operator are approximated via forward differences

$$\delta_x^+ u_{i,j,k} = \frac{u_{i+1,j,k} - u_{i,j,k}}{h_V} \qquad \delta_y^+ u_{i,j,k} = \frac{u_{i,j+1,k} - u_{i,j,k}}{h_V} \qquad \delta_z^+ u_{i,j,k} = \frac{u_{i,j,k+1} - u_{i,j,k}}{h_V} \tag{4.53}$$

In order to ensure the adjointness (Definition 2.8) of the gradient and the divergence operator, i.e. $\nabla^* = -\operatorname{div}$, a similar condition has to be fulfilled in the discrete setting, that is, for every $\boldsymbol{p} \in P$ and $u \in U$ the equation $\langle \nabla u, \boldsymbol{p} \rangle_P = \langle u, -\operatorname{div}(\boldsymbol{p}) \rangle_U$ should hold. This fixes the choice of discretizing the divergence operator $\operatorname{div} : P \to U$ of a vector field $\boldsymbol{p} = (p^1, p^2, p^3)$

$$\operatorname{div} \boldsymbol{p} = \frac{\partial p^1}{\partial x} + \frac{\partial p^2}{\partial y} + \frac{\partial p^3}{\partial z} \tag{4.54}$$

with corresponding backward differences

$$(\operatorname{div} \boldsymbol{p})_{i,j,k} = \delta_x^- p^1_{i,j,k} + \delta_y^- p^2_{i,j,k} + \delta_z^- p^3_{i,j,k} \tag{4.55}$$

being defined as

$$\delta_x^- p_{i,j,k} = \frac{p^1_{i,j,k} - p^1_{i-1,j,k}}{h_V} \qquad \delta_y^- p_{i,j,k} = \frac{p^2_{i,j,k} - p^2_{i,j-1,k}}{h_V} \qquad \delta_z^- p_{i,j,k} = \frac{p^3_{i,j,k} - p^3_{i,j,k-1}}{h_V} . \tag{4.56}$$

This discretization scheme has been used for the gradient descent and primal-dual optimization methods in combination with the corresponding boundary conditions. For solving the linearized Euler-Lagrange Equation (4.23) within the LDFPI-scheme the following discretization has been applied. The divergence operator with diffusivity g is according to its definition

$$\operatorname{div}(g \nabla u) = \frac{\partial}{\partial x}\left(g \cdot \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(g \cdot \frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(g \cdot \frac{\partial u}{\partial z}\right) \tag{4.57}$$

The outer spatial derivates are approximated by means of central differences evaluated at half-grid locations:

$$\frac{\partial}{\partial x}\left(g \cdot \frac{\partial u}{\partial x}\right) \approx \left(g \cdot \frac{\partial u}{\partial x}\right)(x+\frac{1}{2},y,z) - \left(g \cdot \frac{\partial u}{\partial x}\right)(x-\frac{1}{2},y,z) \tag{4.58}$$

$$\frac{\partial}{\partial y}\left(g \cdot \frac{\partial u}{\partial y}\right) \approx \left(g \cdot \frac{\partial u}{\partial y}\right)(x,y+\frac{1}{2},z) - \left(g \cdot \frac{\partial u}{\partial y}\right)(x,y-\frac{1}{2},z) \tag{4.59}$$

$$\frac{\partial}{\partial z}\left(g \cdot \frac{\partial u}{\partial z}\right) \approx \left(g \cdot \frac{\partial u}{\partial z}\right)(x,y,z+\frac{1}{2}) - \left(g \cdot \frac{\partial u}{\partial z}\right)(x,y,z-\frac{1}{2}) \tag{4.60}$$

In turn, the derivatives at half-grid locations are approximated by central derivatives of adjacent pixels and the corresponding diffusivity value is averaged:

$$\left(g \cdot \frac{\partial u}{\partial x}\right)(x+\frac{1}{2},y,z) \approx \frac{g(x+1,y,z)+g(x,y,z)}{2} \cdot (u(x+1,y,z)-u(x,y,z)) \tag{4.61}$$

$$\left(g \cdot \frac{\partial u}{\partial x}\right)(x-\frac{1}{2},y,z) \approx \frac{g(x-1,y,z)+g(x,y,z)}{2} \cdot (u(x,y,z)-u(x-1,y,z)) \tag{4.62}$$

$$\left(g \cdot \frac{\partial u}{\partial y}\right)(x,y+\frac{1}{2},z) \approx \frac{g(x,y+1,z)+g(x,y,z)}{2} \cdot (u(x,y+1,z)-u(x,y,z)) \tag{4.63}$$

$$\left(g \cdot \frac{\partial u}{\partial y}\right)(x,y-\frac{1}{2},z) \approx \frac{g(x,y-1,z)+g(x,y,z)}{2} \cdot (u(x,y,z)-u(x,y-1,z)) \tag{4.64}$$

$$\left(g \cdot \frac{\partial u}{\partial z}\right)(x,y,z+\frac{1}{2}) \approx \frac{g(x,y,z+1)+g(x,y,z)}{2} \cdot (u(x,y,z+1)-u(x,y,z)) \tag{4.65}$$

$$\left(g \cdot \frac{\partial u}{\partial z}\right)(x,y,z-\frac{1}{2}) \approx \frac{g(x,y,z-1)+g(x,y,z)}{2} \cdot (u(x,y,z)-u(x,y,z-1)) \tag{4.66}$$

The advantage of computing derivatives between grid points is that this scheme still only needs the common small 6-neighborhood structure, although second derivates need to be approximated. This makes the scheme highly efficient, especially when implemented on a GPU.

Part II.

Single-View Reconstruction

5. Introduction

A picture is worth a thousand words.

Arthur Brisbane
(Newspaper Editor, 1864 - 1936)

One of the most impressive abilities of human vision is the extraction of three-dimensional information from images. Human observers have an excellent ability to generate plausible $3D$ models of objects around them – even from a single image. To this end, they partially rely on prior knowledge about the geometric structures and primitives in their world. Yet, they also generate plausible models of objects they have never seen before. It is beyond the scope of this work to contemplate on the multitude of criteria the human visual system may be employing for solving the single-view reconstruction problem. Instead, we will demonstrate in this part, that for a large variety of real-world images very simple extremality assumptions give rise to convincing $3D$ models.

From the mathematical point of view, depth information is lost due to the projection. In contrast to multi-view methods, this operation cannot be simply inverted. Hence, depth information can only be guessed by image features like object contours, edges and texture patterns. Especially for images of textured objects under complex lighting conditions, shape from shading methods usually fail to work and further assumptions or user interactions are required.

In this part of the thesis we aim to reconstruct objects from single real-world photographs taken under arbitrary light conditions and containing objects with arbitrary topology and texture. An example for single-view reconstruction is shown in Figure 5.1. We aim to quickly generate plausible 3D models from single images and tackle the ill-posedness of the problem with simple priors, and user input which we aim to keep as low as possible. To further simplify the problem, we do not aim for an accurate metric 3D reconstruction. In many cases such as photo or video editing, it often suffices to have an approximate shape of the object in question.

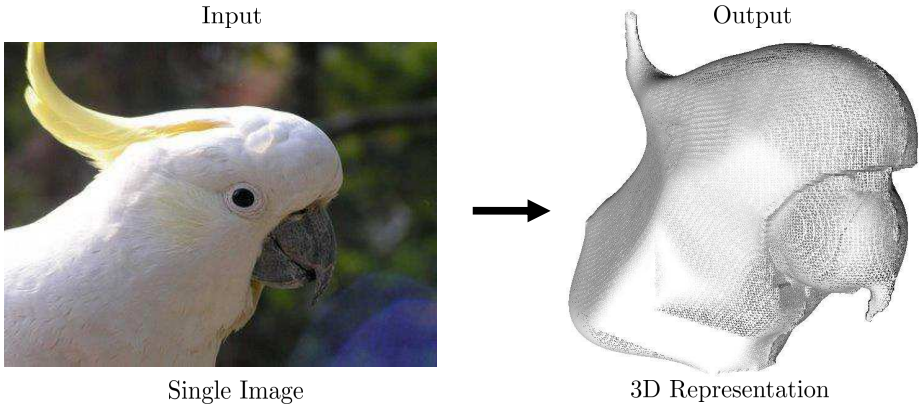


Figure 5.1.: Illustration of single-view reconstruction task. The goal is to obtain 3D geometry from a single input image.

Before we define our problem setting and detail our approach in Section 5.4, we give an overview of related work in the following section and classify them with respect to their major properties.

5.1. Related Work and Classification of Single-View Reconstruction Algorithms

In this section we give a survey on the subject of single-view reconstruction. We provide an introduction to the field and examine basic image information and assumptions that are used in order to compensate for ill-posedness. In the next section we then review, categorize and compare existing state-of-the-art approaches.

For specific assumptions imposed on an image a variety of methods to estimate 3D geometry exist in literature. However, a thorough comparison has not been carried out so far.

The reason for this lies partly in the significant diversity of existing approaches which in turn is due to the inherent ill-posedness of the underlying problem: during image formation, depth is irrecoverably lost. In their effort to make the problem tractable, single-view methods have come up with an abundance of very different assumptions, methods and priors to infer the geometry of a depicted scene or object. The reconstruction precision of such approaches exceeds that of plausible estimates only in very few cases. Consequently, the reconstruction objectives are of very different nature, which makes a comparison difficult.

The geometric information that is to be retrieved from a single image can be of very different manifestation reaching from purely relational information, sparse metrics or dense depth information to a complete 3D model of a single object or even a scene. This circumstance in combination with the inherent ill-posedness of the problem is the main reason for the strong diversity that is witnessed among the works in single-view reconstruction and it is by no means a straightforward task to develop a taxonomy let alone a comparison.

In the following we will give an overview on the different types of image information ("*image cues*") used in the different reconstruction processes and list typical *priors* that are assumed in order to overcome the ill-posedness. This will also serve as a survey on related single-view works. Later in Section 5.2, we will classify a number of single-view approaches and compare their properties.

5.1.1. Image Cues

Approaches to single-view reconstruction extract specific higher or lower level information contained in the input image either automatically or with the help of user input. This information is then interpreted to infer geometric relations of the depicted object or scene. In the following we list the most important categories and give prominent references.

Shading. The problem of *Shape from Shading* (SfS) is to infer a surface (height field) from a single gray level image by using the gradual variation of shading that is induced by the surface interaction of light. Some approaches also co-estimate lighting conditions and reflection properties. In general, the goal is to find a solution to the following image formation model

$$R(n(x)) = I(x) \ , \tag{5.1}$$

where I is the image, n is the normal field of the surface and R is the reflectance function which is dependent on the object. In most SfS approaches a Lambertian reflectance model

is assumed. There are, however, other models which also include the specular case (e.g. Wang et al. [235]). SfS is an ill-posed problem, although there has been progress on deriving conditions for unique solutions by Prados and Faugeras [180].

As shown by Durou et al. [75] and Zhang et al. [254] reconstruction from real world images is limited in the sense that each approach exhibits special and sometimes unrealistic requirements on lighting conditions or reflection properties. Especially the presence of texture is an issue. Work has been done, however, to incorporate interreflection [162], shadowing and perspective projection [61] just to name a few. One of the first minimization approaches to SfS is by Ikeuchi and Horn [120]. For a current survey see Durou et al. [75].

Shadow. The shadow that is thrown by objects conveys geometric information relative to the viewpoint of the light source. This information can be used to narrow down possible reconstruction results. Often point light sources have to be assumed as soft shadows do not provide enough information. Furthermore, shadow is not always thrown on known geometry, which makes the problem even more complex. Apart from reconstruction ambiguities, it is not straightforward to extract shadow borders from images. References include works by Daum and Dudek [67], Kender and Smith [131], Yu and Chang [246] and Hatzitheodorou [109].

Contour Edges. Contour edges are salient structures in the image that are induced by surface discontinuities, occlusion, object boundaries or reflectance changes. They give evidence for geometry and relative pose/position of objects. *Junction points* or *corners*, where multiple contour edges meet or end, are also of importance for single-view reconstruction.

Subclasses of contour edge-based methods are contour-based and silhouette-based reconstruction methods. *Shape from Contour* approaches try to infer geometry from given or estimated object contours. With contour, we refer to the set of all visible points on a surface, whose image rays are tangent to the surface. In most cases reconstructions are ambiguous, especially smooth surfaces often do not exhibit sufficient contour lines in the image. Shape from Contour approaches based on closed contour drawings include Horaud et al. [112], Ulupinar et al. [214] and Li et al. [151]. Karpenko et al. [127, 126] interpret user line drawings. Other single-view reconstruction approaches that use contour edges for reconstruction include [70, 104, 138, 110, 192, 193].

Silhouette. Closely related to Shape from Contour are approaches that infer geometry given the object silhouette. Such as in a shadow play, the silhouette of an object is defined as the set of all points in the image plane being covered by the projection of the objects' surface onto the image plane. Thus, the silhouette forms a solid shape with a featureless interior and with a single closed contour that corresponds to the outline of the object.

The goal of silhouette based approaches is to find a geometric reconstruction, whose projection into the image plane agrees with the silhouette. As there are always infinitely many objects that are silhouette consistent this cue suffers from inherent ambiguity if used alone.

There are several silhouette based single-view reconstruction algorithms that we will consider in more detail later. These include works by Prasad et al. [182, 183], Oswald et al. [1] and Töppe et al. [3]. Related to these approaches are a class of sketch based modeling tools e.g. by Igarashi et al. [117], Karpenko et al. [126] and Nealen et al. [163].

Texture. Besides geometry, the appearance of real world objects is also determined by texture. Although a complete distinction from shading is not possible, texture is considered

as an inherent property of an object rather than a result of an interaction of light and geometry.

If one assumes objects to have a regular and known texture it is possible to infer their geometry from the way the texture is deformed after image projection. These *Shape from Texture* approaches, obviously, impose strong constraints on the reconstructable objects. An example constitutes the work of Malik and Rosenholtz [158].

Further single-view reconstruction algorithms that use texture cues include Super et al. [208], Hassner and Basri [108] and Vetter et al. [226]. Approaches that combine texture and contour edges for reconstruction by considering so-called 'superpixels' are Hoiem et al. [110] and Saxena et al. [193].

Defocus. Due to physical aspects of image formation, the sharpness of a depicted object correlates with its distance to the camera. This fact can be used to infer a dense depth map from an image. However, the accuracy of such methods is limited and camera calibration is necessary. References include works from Levin [149] and Bae and Durand [16].

Location. The location of objects in the image can infer semantic knowledge of the objects. For example, ground, floor or sky can be identified more easily from their location in the image. This information can be helpful for 3D reconstructions. Hoiem et al. [110] reconstruct vertical objects by distinguishing them from the ground and the sky. Delage et al. [70] use a Bayesian network to identify floor pixels.

5.1.2. Priors

Priors are of utter importance in single-view reconstruction in order to compensate for the problem of ill-posedness. Depending on the ultimate reconstruction goal and the target group of objects, different priors or a combination of them can be applied. Priors can either be chosen in fixed form, or they can be gained by learning. Furthermore, there are low-level and high-level priors. In the following we will list priors that are most frequently assumed in single-view reconstruction.

Smoothness. Smoothness can be defined as the small spatial change of some property. In single-view reconstruction we are often not able to infer a dense reconstruction. It is therefore good practice to choose among the possible reconstruction surfaces those which tend to be smooth. Smoothness naturally plays a significant role in the reconstruction of curved surfaces as in [253, 183],[1, 3].

Smoothness in a wider sense can also be learned as the consistency of object surfaces. Hoiem et al. [110] use a machine learning approach to find image features indicating the assignment of neighboring superpixels to the same object. Saxena et al. [193] use image cues and geometric relations to learn the relative depth of neighboring superpixels. Liu et al. [154] use a semantic segmentation of the image and infer a depth value for each pixel based on the predicted semantic label.

Geometric Relations. Basic geometric relations are often encountered specifically in man-made environments. As a prior they can help to dissolve ambiguities in the reconstruction process. As part of those basic geometric relations we consider e.g. *coplanarity*, *collinearity*, *perpendicularity* and *symmetry*. An early work which makes use of such simple rules is the one of Lowe [157]. By assuming planes to be parallel or perpendicular one can also derive camera

parameters (see Criminisi et al. [63]). This is even more important, as perspective projection is not angle-preserving and the image of parallel lines will not necessarily be parallel. We can often assume objects to stand vertically on the ground plane [110, 70, 104], or can infer depth relations from parallel lines or curves [140].

Symmetric objects exhibit identical sections, which are projected to different locations in the input image. Knowing that these parts possess similar geometric and reflectance properties one can interpret their respective projections as views of the same object part from different observation points. This can be seen as a weak multi-view scenario providing more information for reconstruction [111]. Also, occluded geometry can be inferred from this prior [104].

Volume / Area. With smoothness as a prior on its own, solutions tend to be trivial or flat depending on the reconstruction approach. Adding a volume prior to the reconstruction process will ensure an inflation of the object surface and will still result in a certain compactness of the solution due to the smoothness assumption. Volume priors can be found in Li et al. [151] and Töppe et al. [3].

Semantic Relations. Semantic relations infer high-level knowledge on the relative position and inner structure of different objects and their depth values. Han and Zhu [104], for example, infer occluded points based on semantic human knowledge, e.g. that leaves are connected to the plant. Koutsourakis et al. [138] introduce semantic knowledge to ensure the consistency of different floors. Finally, knowledge on the location of the ground and the sky represents an important cue for 3D reconstruction. The ground is often used as starting point for the reconstruction as objects, especially walls, are usually perpendicular to this plane [104, 70, 110].

Shape Priors. Shape priors impose high-level knowledge on the objects to be reconstructed. Among commonly used priors, full shape priors usually impose the strongest restrictions. On the one hand, this leads to a rather limited applicability of the approach. On the other hand, the reconstructions are usually of high quality and work automatically without user input.

Shape priors can be defined or learned. In [138], Koutsourakis et al. define a full shape grammar for the reconstruction of facades. This limits the approach to the reconstruction of buildings in urban environments. In contrast, Rother and Sapiro [190] and Chen and Cipolla [55] shape priors are learned from a database of sample objects. Hence, they are not a-priori limited to a specific object class. However, their choice of samples intrinsically limits their approach to the object classes represented in the database.

The representation of shape priors ranges from specified sets of grammar rules over parametric models to probabilistic priors. In [55], Chen and Cipolla learn depth maps of human bodies by means of principal component analysis. This model imposes strong assumptions on the 3D object, but the dimension of the state space is reduced and only valid 3D reconstructions are obtained. In contrast, Rother and Sapiro [190] impose less strong assumptions on the learned model. For each object class a shape prior is learned as the relative occupancy frequency of each voxel in the object.

5.2. Classification of Single-View Approaches

In this section we will examine selected works in the field of single-view reconstruction. Due to the abundance and diversity of approaches we selected algorithms with respect to the

following criteria: the chosen approaches are applicable to real world images that are not taken under special or unrealistic material or lighting conditions. We rather concentrate on approaches inferring objects from ordinary photographs where reconstruction plausibility is more important than precision. The selection, furthermore, focuses on works that are representative and state-of-the-art. We provide a classification and examine differences and similarities.

For classification we found several categories ranging from application domain over image cues and shape priors to user input and the surface representation. However, these categories are not suitable to partition the set of approaches due to strong overlap. Instead, we think that the most important information for each single-view reconstruction approach is its application domain, i.e. the set of objects and scenes, which can be reconstructed. We introduce the literature sorted by the following four categories:

- Curved Objects
- Piecewise Planar Objects
- Learning Specific Objects
- 3D Impression from Scenes

We distinguish between objects and scenes. Reconstructions of scenes aim at producing 3D impressions or depth maps from the whole scene contained in the image. In contrast, object reconstruction approaches focus on single objects within the scene. Approaches that reconstruct *curved objects* principally aim at producing arbitrary, mostly smooth objects. Often, minimal surface approaches are used, which try to minimize the surface of the object given a fixed area or volume. The second class consists of methods that focus on *piecewise planar objects* such as buildings and man-made environments. Furthermore, we distinguish arbitrary curved and planar objects from *learning specific objects*. Approaches in this class cannot reconstruct arbitrary objects, but are inherently limited to specific object classes by shape information learned from sample databases. Finally, we discuss methods that do not aim to reconstruct exact or plausible 3D geometry but rather provide a pleasing *3D Impression from Scenes*. In the following, we will present and classify approaches to related single-view approaches.

5.2.1. Curved Objects

In this category we list works which aim to reconstruct object with a smooth curved surface. Since we also aim for reconstructing curved objects, this category will list competing methods to which we later compare and which we therefore describe in more detail.

Zhang et al. Zhang et al. [253] proposed a method for interactive depth map editing based on an input image. The depth map reconstruction is the result of minimizing a thin plate energy [74], which favors smooth surfaces and penalizes bending. User input comes as a variety of constraints on the thin plate energy that can be placed interactively into the depth map. These comprise of position constraints, surface normals, surface or normal discontinuities, planar region constraints or curves on which curvature or torsion is minimized.

From a mathematical point of view the thin plate energy for a continuous function f on a two dimensional rectangular domain $[0,1]^2$ is defined as:

$$E(f) = \int_0^1 \int_0^1 \left[\alpha(u,v) \left\| \frac{\partial^2 f}{\partial u^2} \right\|^2 + 2\beta(u,v) \left\| \frac{\partial^2 f}{\partial uv} \right\|^2 + \gamma(u,v) \left\| \frac{\partial^2 f}{\partial v^2} \right\|^2 \right] du\, dv \ , \tag{5.2}$$

where functions $\alpha, \beta, \gamma : [0,1]^2 \to \{0,1\}$ extend the thin plate model with weighting functions which can be used to define local surface discontinuities. Zhang et al. [253] discretize this minimization problem by introducing a function $g_{i,j}$ that samples values of the depth map function $f : [0,1]^2 \to \mathbb{R}$ on a discrete rectangular grid, that is, $g_{i,j} = f(id, jd)$, with d being the distance between neighboring grid points. For efficiency and accuracy the grid resolution can be locally refined by the user. By stacking all values $g_{i,j}$ into a single vector $\mathbf{g}$ and by discretizing the partial derivatives of g, the energy in Equation (5.2) can be written in matrix form as

$$\mathbf{g}^T\mathbf{C}\mathbf{g} \qquad \text{subject to} \quad \mathbf{A}\mathbf{g} = \mathbf{b} \ , \tag{5.3}$$

where the condition $\mathbf{A}\mathbf{g} = \mathbf{b}$ may contain any constraints on the surface that can be expressed in linear form. For a detailed description on how the constraints are incorporated into this quadratic optimization problem we refer to [253]. A description of these constraints from the user's point of view is given later together with the experimental comparison (Section 9.1.2).

Prasad et al. The works [183] and [182] of Prasad et al. introduce a framework for single-view reconstruction of curved surfaces. The method is related to the one by Zhang et al. [253] but aims at reconstructing closed surfaces.

The main idea involves computing a parametric minimal surface by globally minimizing the same thin plate energy (Equation (5.2)) as Zhang et al. [253], with the difference, that they minimize with respect to a parametrized 3D surface $f : [0,1]^2 \to \mathbb{R}^3$ instead of a depth map. As a result, function domain and image domain are no longer equivalent. For implementation purposes, the discretization of the optimization problem with constraints is done similar to Zhang et al. [253] (see Equation (5.3)).

The choice of constraints is mostly different from Zhang et al. [253]. The main source of reconstruction information is the silhouette: Prasad et al. [183] use the fact that normals along the contour generator $c(t)$ can be inferred from the 2D silhouette as by definition they are parallel to the viewing plane for a smooth surface. This leads to the constraints

$$\pi(f(u(t), v(t))) = c(t) \tag{5.4}$$

$$n(c(t))f(u(t), v(t)) = 0 \qquad \forall t \in [0,1] \ , \tag{5.5}$$

where $n(c(t))$ is the normal at the point $c(t)$ in $\mathbb{R}^3$ and π the orthographic projection function. The user has to determine the coordinates $(u(t), v(t))$ of the contour generator in parameter space. This is done by placing lines within the grid of the parameter space and setting them in correspondence with the parts of the contour generator. Similar to Zhang et al. [253] the user can employ position constraints to define the object inflation locally. Also, surface discontinuities can be optionally specified to relax the surface smoothness along curves in the parameter space.

Important to note is that in order to define the topology of the object, the user has to define which parts of the parameter space boundary are connected. For example, the connection of the left and right boundary defines a cylindrical shape of the function domain.

Other Approaches. Francois and Medioni [87] present an interactive 3D reconstruction method based on user labeled edges and curves, which are represented by non-uniform rational basis splines (NURBS). The reconstructed objects are either modeled as generalized cylinders or as a set of 3D surfaces. Terzopoulos et al. [209] propose deformable elastic 3D shape models, which evolve around a symmetry axis and whose projection into the image is attracted by strong image gradients. Cohen and Cohen [56] propose a generalization of snakes to 3D objects based on a sequence of 2D contour models for medical images.

Another approach to 3D reconstruction are surfaces of revolution [241, 221, 57]. They are common in man-made objects and represent a subclass of straight homogeneous generalized cylinders. These approaches strongly rely on the assumption of rotational symmetry of the objects. A surface of revolution is obtained by revolving a planar curve, referred to as scaling function, around a straight axis, the revolution axis. For instance, Colombo et al. [57] formulated the task of 3D reconstruction as the problem of determining the meridian curve from the imaged object silhouette and two given imaged cross sections. Based on the computation of fixed entities such as the vanishing line or the objects' symmetry axis, camera calibration can be done and the surface of revolution is inferred. Texture acquisition is obtained by inverse normal cylindrical projection.

A closely related work that appeared after the publication of our approach is by Chen et al. [54]. Based on basic one input the approach fit generalized cylinders or cuboids into an input image which automatically align with prominent edges in the image and texture information is automatically transferred. Several of these simple objects can be combined to create more sophisticated 3D models. This work can be seen as a generalization of the work by Terzopoulos et al. [209].

5.2.2. Piecewise Planar Objects and Scenes

In this category we specify related single-view approaches that aim to reconstruct piecewise planar surfaces as can be found in many man-made environments. Some methods even restrict the orientation between surfaces to be either parallel or orthogonal. Generally, methods in this category are not able to reconstruct smooth, curved surfaces.

Kanade [125] recovers shape from geometric assumptions. The world is modeled as a collection of plane surfaces, which allows for a qualitative object recovery. Quantitative recovery is achieved by mapping image regularities into shape constraints. Piecewise planar scenes are computed by Liebowitz et al. [153] based on camera and geometric constraints such as parallelism and orthogonality, e.g. for the reconstruction of buildings.

Criminisi et al. [63] describe how 3D affine measurements can be obtained from a single image that depicts planes and parallel lines under perspective projection. They estimate vanishing points and lines in order to compute distances between parallel planes and lines which finally enables them to compute a basic 3D model of the scene.

Delage et al. [70] describe an approach for the automatic reconstruction of 3D indoor scenes that only contain orthogonal planes ("Manhattan world assumption"). Assuming the camera calibration to be known, the authors estimate plane and edge orientation by means of a Markov Random Field (MRF) [150]. The work can be seen as a modification and generalization of Sturm and Maybank [206].

In [138], Koutsourakis et al. generate urban 3D reconstructions from images by estimating the parameters of a 3D shape grammar in a MRF approach, so that the generated building best matches the image. Due to the shape grammar the approach always produces well-defined buildings and the complexity of the optimization as well as the dimensionality of the problem is strongly reduced.

Apart from symmetry and planarity, two additional shape constraints for object reconstruction are introduced by Li et al. [151]: maximum compactness and minimum surface. Instead of computing vanishing lines, Kushal et al. [139] perform 3D reconstruction of structured scenes by registering two user indicated world planes. In a later work Kushal and Seitz [140] compute 3D models from vanishing points, line directions and face normals estimated from a single image. Focusing on parallel lines in images of man-made architecture, Ramalingam and Brand [184] recover their corresponding 3D location via linear programming approach with

connectivity constraints. Hong et al. [111] study the relation between symmetry of objects and the viewer's relative pose to the object. An important principle for the reconstruction of symmetric objects is that one image of a symmetric object is equivalent to multiple images. Li et al. [152] describe a method for reconstructing piecewise planar objects by using connectivity and perspective symmetry of objects.

5.2.3. Learning Specific Objects

In this category we list approaches which learn the shape or regularity properties of objects or relations between objects and their appearance.

Han and Zhu [104] propose a 3D reconstruction approach based on manually defined shape priors, which can on the one hand be applied to polyhedral objects and on the other hand to grass and tree-like objects. They represent the 3D scene by two graphs, one consisting 3D objects, the other representing the relations between the objects in the scene. The model is formulated in a Bayesian approach and optimized with Markov Chain Monte Carlo methods.

Rother and Sapiro [190] present a framework for pose estimation, 2D segmentation, object recognition and 3D reconstruction from a single image which is well-suited to reconstruct bounded objects, but not for elaborate scenes. They learn object shapes by means of voxel occupancy grids and perform the reconstruction task as a probabilistic recognition of object pose and class. The most likely hypothesis is computed with a branch and bound algorithm.

Chen and Cipolla [55] propose to infer 3D information directly from shape priors which are learned from pairs of silhouettes and corresponding depth maps. Both silhouettes and depth maps are aligned and dimensionality-reduced by principal component analysis (PCA) and then learned with Gaussian processes. Any given input silhouette is then similarly transformed via PCA and the most likely depth estimate for each pixel is estimated via the trained model.

Hassner and Basri [108] aim at depth reconstruction from a single image based on examples. The samples are given in a database containing mappings of images to their corresponding depth maps. For an input image depth values are inferred by comparing image patches with patches in the database and selecting the most likely one. Different depth hypotheses from overlapping patches are averaged to obtain depth value for single pixels. To ensure consistency of neighboring patches a global optimization procedure is proposed which iteratively refines depth estimates.

Vetter [226] learned a parametric model for the reconstruction of faces by applying PCA to a database of registered 3D faces. Then the model parameters can be found, which best explain the given image of a face. In Nagai et al. [161], objects are learned from a sample database. A Hidden Markov Model is used to model the correspondence between intensity and depth.

5.2.4. 3D Impression from Scenes

In [110], Hoiem et al. propose a fully automatic approach for creating 3D models from single photographs, which is similar to the creating of pop-up illustrations in children's books. They divide the world into ground, sky and vertical objects. The appearance of these classes is described by image cues, which are learned from sample images. An input image is segmented into superpixels. In a probabilistic framework the superpixels are grouped into constellations with similar class labels which are inferred by the trained model. Finally, a depth impression of the picture is obtained by aligning class label-specific image parts in 3D.

Saxena et al. [193] propose another approach for obtaining 3D structure from a single image of an unstructured environment. The only assumption the authors make is that the world

consists of small planes, whose 3D position and orientation is to be estimated. Similar to Hoiem et al. [110], the authors start out from a superpixel segmentation of the image. Then they train a Markov random field (MRF) model to learn the relation between superpixel appearance and its corresponding depth and orientation. A polygonal mesh representation of a scene is obtained via maximum-a-posteriori inference of the trained MRF-model.

In Horry et al. [114], simple 3D scenes are reconstructed based on user input such as vanishing points and foreground objects. The background of the scene is then modeled by rectangles, the foreground by hierarchical polygons. Barinova et al. [18] propose a reconstruction approach for urban scenes yielding visually pleasing results. The method is based on fitting 3D models containing vertical walls and ground plane to the scene.

In the next section we give an overview of selected methods from each category and compare them with respect to their properties.

5.3. Properties and Comparison of Related Works

In this section, we compare important works of each category with respect to several properties: image cues, priors, surface representation, important assumptions, type of user input and the method's precision. Table 5.1 compares the presented approaches and the ones presented in this thesis (Chapters 6 to 8) with respect to their categories and properties.

Category, Assumptions, Precision. As described above we grouped the related work into four categories which reflect their application domain (first column of Table 5.1).

The applicability of an approach is also characterized by its **assumptions**. If specific assumptions are not met, the reconstruction process easily fails. Assumptions for each method are given in column five of Table 5.1. Typical assumptions are a calibrated camera [70], a simplified scene composition [70, 110], an object database containing samples for learning shape priors [55, 190], a specific viewpoint [183],[1, 3] or given geometric properties such as vanishing lines of reference planes [63].

Another aspect which determines the applicability of an approach to a special problem is its envisaged reconstruction **precision**. The precision of a method describes the consistency of the reconstructed 3D model with the actual real-world scene. There is a trade-off between precision and reconstruction feasibility. One can witness a correlation between reconstruction precision and requirements: the higher the envisaged reconstruction precision, the more assumptions and priors have to be made on the reconstruction domain.

Reconstructions can be exact, if the computed lengths and orientations of the inferred 3D objects accurately correspond to the true object. This is usually only possible from a single image if strong assumptions are made, e.g. piecewise planarity with only three orientations (Manhattan assumption) [70] or known reference heights and a calibrated camera [63]. Since such strict assumptions strongly limit the applicability of the approach, most approaches revert to computing the most likely solution to the ill-posed reconstruction problem without guaranteeing accuracy. The probability of a solution is usually measured by means of manually defined [104] or learned shape priors [55, 190]. We call this a *plausible* precision. Finally, there are approaches, which do not aim at reconstructing the real object. Instead, they find solutions which look good to the viewer when animated [110, 193, 114] or can be used to synthesize approximate new views of a scene. We call these reconstructions *pleasing*. The reconstruction precision is indicated in the third column of Table 5.1. '=' indicates exact precision, '$\simeq$' plausible precision and '$\approx$' a pleasing approach. Surely there are smooth transitions between these classes.

Category	Method	Precision	Surface Representation	Assumptions	User Input	Image Cues				Priors				
						Silhouette	Edges	Location	Texture	Smoothness	Volume or Area	Semantic Relation	Geom. Relation	Shape
Curved Objects	Prasad et al. [183]	$\simeq$	[closed] parametric	charateristic sideview, max. genus 2	contours, [creases]	x			x	x				
	Zhang et al. [253]	$\simeq$	depth map	none	constraints					x				
	Shape Prior proposed in Chapter 6/[1]	$\simeq$	closed implicit	sideview, symmetry	silhouette, [creases], [data term]	x				x				
	Vol.Prior 3D proposed in Chapter 7/[3]	$\simeq$	closed implicit	sideview, symmetry	silhouette, [creases], [volume]	x				x	x			
	Vol.Prior 2.5D proposed in Chapter 8/[5]	$\simeq$	depth map	sideview, symmetry	silhouette, [creases], [volume]	x				x	x			
	Colombo et al. [57]	$\simeq$	closed parametric	rotational symmetry	silhouette, cross sec.	x							x	
Piecewise Planar	Criminisi et al. [63]	=	non-closed polygonal	vanishing line, refer. height, ground plane	all edges to be measured		x						x	
	Delage et al. [70]	$\simeq$	non-closed polygonal	calibration, Manhattan	none		x	x	x	x		x	x	
	Koutsourakis et al. [138]	=	closed polygonal	rectified image, buildings	none		x		x			x		shape grammar
Learning Specific Objects	Han & Zhu [104]	$\simeq$	closed polygonal	polyhedra, plants	none		x			x		x	L	proba-bilistic
	Rother & Sapiro [190]	$\simeq$	closed implicit	calibration, color models, database	none			x						learned voxel model
	Chen & Cipolla [55]	$\simeq$	depth map	database	silhouette	x								learned PCA model
	Hassner & Basri [108]	$\simeq$	depth map	fixed view, spec. object database	none				x	x				learned example based
Scenes	Hoiem et al. [110]	$\approx$	pw. planar depth map	simple scene: sky, vertical walls&ground	none		x	x	x	L		x		
	Saxena et al. [193]	$\approx$	pw. planar depth map	world consists of planes	none		x	x	x	L			x	

Table 5.1.: Overview of single-view methods: for each approach the most important characteristics are indicated: Precision of the method (exact '=', plausible '$\simeq$', pleasing '$\approx$'), the representation of the 3D object, important assumptions made by the approach, the necessary user input and image cues as well as priors which are used in the reconstruction process. The 'L' indicates a prior which is not assumed but learned by the approach. Terms in brackets are optional.

Surface Representation. The form of surface representation is closely connected to the reconstruction algorithm. Firstly, only those objects are reconstructable that can be adequately represented. Seen the other way, the representation has to reflect the reconstruction domain well. And secondly, the representation model has to conform to the reconstruction process.

We distinguish between parametric and implicit surface representations. Each point on a *parametric surface* can be uniquely described by a coordinate. Finding a good parametrization for an object is not straightforward and generally does not allow for arbitrary topology. *Implicit surfaces* are a remedy to this problem. In this case, the surface is a level set of a function defined on $\mathbb{R}^3$. In contrast to parametric surfaces, single points on the surface are not easily addressed. *Polygonal* surface representations are neither parametric nor implicit and can be described as a planar graph with nodes, edges and faces. Note that polygonal surfaces often describe piecewise planar objects but are also used for approximating curved parametric surfaces. Finally, representations can describe *closed* and *non-closed* 3D surfaces. As a special case we also regard *depth maps*, which assign a depth to each pixel.

User Input. Completely automatic reconstruction on a single input image is often not feasible or the output quality is limited. Therefore, the user may be required to give cues on important image features. Most single-view approaches aim to keep user input simple. User input can convey low-level and high-level information. High-level input is of semantic quality which helps to dissolve ambiguities, e.g. the object silhouette.

This stands in contrast to tools, where the user *models* the reconstruction with the help of low-level operations, e.g. by specifying surface normals or cutting away object parts. Many of these *modeling tools* [123, 240, 24] are not image-based and therefore only remotely related to single-view reconstruction. In *Sketch-based modeling tools* [117, 163, 126, 249] such modeling operations are driven by user indicated lines. The Teddy tool will be examined in more detail in Section 9.1. A pioneering work on free-form modeling was done by Welch and Witkin [238].

There is 2D and 3D user input. Most approaches use 2D input which in most cases is directly applied to the input image [1, 3, 5]. This involves tracing contour edges such as creases or vanishing lines. 3D input is directly applied to the reconstruction surface and is often more involved for the user as he needs to navigate in 3D space (e.g. specifying normals).

For some approaches the user input stage is separate from the reconstruction stage [55, 63]. Other methods compute a first reconstruction, then the user can add further input and the process is continued, e.g. [253, 183] and our proposed approaches in Chapters 6 to 8 [1, 3, 5].

Image Cues and Priors. The last columns of Table 5.1 list image cues and priors that have been identified previously in Sections 5.1.1 and 5.1.2. A cross "x" in the respective column indicates that a methods makes use of this image cue or prior, while an "L" indicates that the respective prior has been learned.

5.4. Problem Setting and Approach

In this section we specify the problem setting of our single-view reconstruction approach, explain its central idea and crucial assumptions, describe the general workflow and mention possible applications.

5.4.1. Problem Statement

As revealed in the previous three sections, so far every approach to single-view reconstruction focuses on a particular subproblem, e.g. the class of scenes or input images, in order to diminish the ill-posedness of the reconstruction task. Currently, there exists no method which is able to deal with the complexity of the general single-view reconstruction problem. Our work will not be an exception. We will also restrict the problem class and rely on a small amount of user input. Our setup pursues the following *goals* for the reconstruction of objects from a single image:

- Arbitrary object topology (with respect to its silhouette).
- Arbitrary light conditions.
- Arbitrary object materials.
- Realistic and non-realistic images (e.g. photos and paintings).
- Full 3D object reconstructions (rather than depth maps).
- Amount of user input is small/minimal.

These goals can be achieved by restricting the class of objects and making the following *assumptions*:

- Plausible rather than exact 3D reconstructions.
- The exact silhouette of the object in the image is provided.
- The topological genus of the object is equal to the one of its silhouette.
- The input image is a side view of a plane symmetric object.
- We aim for curved objects with a mostly smooth surface.

In order to achieve these goals we propose the following approach to single-view reconstruction.

5.4.2. Our Approach to Single-View Reconstruction

In our approach we heavily rely on the expressiveness of *object silhouettes*, which can usually be simply extracted with recent segmentation techniques. Further, we will make use of the minimum surface approach explained previously in Chapter 3. That is, we are aiming to compute a minimal surface that is consistent with a given input silhouette. Clearly, these two ingredients are not sufficient to create three-dimensionals objects, since a silhouette-consistent minimal surface is entirely flat. Therefore, additional constraints are necessary to *inflate* the objects into the third dimension. In short, we propose silhouette-consistent minimal surfaces in combination with an inflation heuristic for reconstruction of objects from a single image.

Intuitively, the main idea of our approach can be imagined as inflating a soap bubble (like the one in Figure 7.2) that spans the frame of arbitrary shape which is given by the silhouette contour of some object.

In this part of the thesis, we propose two different inflation techniques which are appropriate for this task and reflect our goal of minimizing the amount of user input. We also explain how these approaches can be efficiently computed. Moreover, we will show on a variety of examples that our approach targets a limited but relevant class of real world objects.

Applications. Our approach is particularly applicable to image and video editing, e.g. when copying an object from one image to another. With our approach novel views on the object

can be synthesized. Once having a 3D representation of an object one can change its material and reflectance properties. One can perform three-dimensional relighting of the object, e.g. to match the light conditions in the new image after copying. Further, our approach is a powerful tool for any image-based modeling. The reconstructions by our algorithms which are quickly obtained from images can be the basis for professional 3D modeling of objects.

5.4.3. Workflow of Our Approach

Image Segmentation. The main prerequisite for a good result with our approach is a reasonable silhouette. The number of holes in the segmentation of the target object determines the topology of the reconstructed surface. Notably, our reconstruction method can also cope with disconnected regions of the object silhouette. A silhouette $S : \Omega \to \{0, 1\}$, being a binary function on the image domain Ω, can be obtained with any segmentation algorithm like the interactive methods in [28, 29, 189, 219]. We use a graph cut-based algorithm which calculates two distinct regions based on respective color histograms which are defined by representational pen strokes given by the user (see Figure 5.2). Based on the silhouette ob-

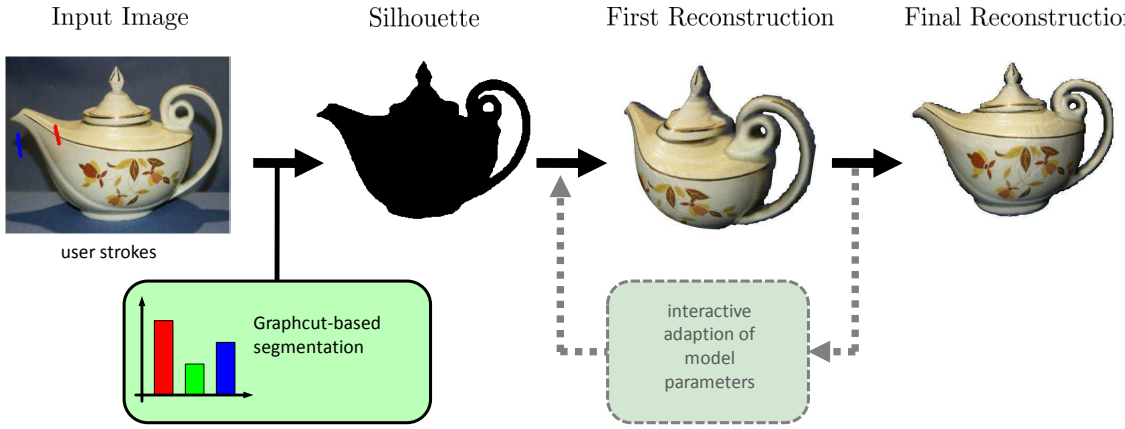


Figure 5.2.: General workflow of all proposed single-view reconstruction approaches. With scribble-based interactive segmentation a silhouette is extracted. A first reconstruction is obtained and can be refined with additional user-input until the reconstruction result is satisfactory.

tained from the input image, our approach inflates the silhouette and produces a first 3D reconstruction (third picture in Figure 5.2). The user can then interactively adapt the model parameters until it suits the needs of the user and eventually obtains the final reconstruction (last picture in Figure 5.2).

5.5. Conclusion

This chapter introduced the problem of single-view reconstruction and gave a detailed overview of the works which are most related to our problem setting. At the same time we provided a classification of single-view approaches into four classes: curved objects, piecewise planar objects, learning specific objects, and 3D impression from scenes. We have identified several properties that help to compare the algorithms, namely: the type of surface representation, method assumptions, type and amount of user input, precision of the output, as well as typical image cues and priors. Further, we defined our considered problem setting and outlined our general approach. In the following three chapters we will discuss particular methods for single-view reconstruction based on minimal surfaces which are combined with different priors and surface representations.

6. Single-View 3D Reconstruction with a Shape Prior

We become what we behold. We shape our tools, and thereafter our tools shape us.

Marshall McLuhan
(Canadian Philosopher, 1911-1980)

6.1. Introduction

In this chapter, we propose a variational convex optimization approach to user-guided 3D reconstruction from a single image. Figure 6.1 gives two examples for input and output of our method. We make use of the previously explained minimal surface prior (Chapter 3) and combine it with a novel silhouette-based shape prior in order to estimate 3D information from a single image. The algorithm targets a limited but relevant class of real world objects.

Contributions. Our single-view reconstruction approach has several desirable properties and makes the following contributions with respect to existing work:

- We present the first approach to single-view reconstruction with a *non-parametric* surface representation. In contrast to existing work, our method can deal with any surface topology and genus due to the proposed implicit surface representation.
- We propose a novel shape prior based on the distance transform of the silhouette that provides a good inflation heuristic for many natural and man-made objects.
- Compared to other state-of-the-art methods our approach needs significantly less user input in order to obtain comparable reconstructions.
- The approach can be solved efficiently in a globally optimal manner and is hence independent of the initialization. Due to parallelization, results can be computed in interactive rates on consumer hardware.

In the following, we will introduce a variational framework for single-view reconstruction and show how it can be solved by convex relaxation techniques. In Section 6.3, we give an overview of the proposed reconstruction framework and explain how users can provide silhouette and

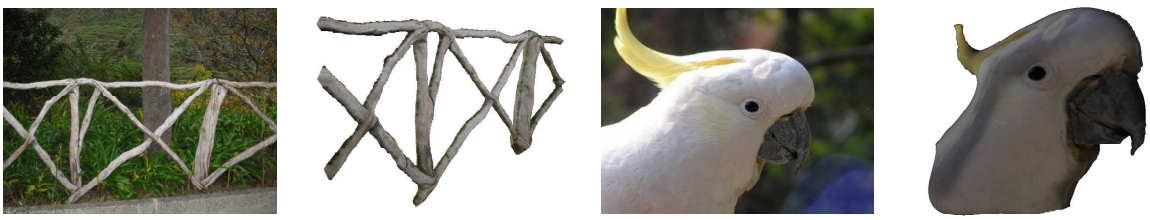


Figure 6.1.: Input images and textured reconstruction results from the method proposed in this chapter.

additional information with minimal user interaction. The viability of our approach is tested on several examples in Section 6.4, followed by concluding remarks in Section 6.5.

6.2. Variational Framework for Single-View Reconstruction

6.2.1. Variational Formulation

Let $V \subset \mathbb{R}^3$ be a volume surrounding the input image $I : \Omega \to \mathbb{R}^3$ with image plane $\Omega \subset V$. We are looking for a closed surface $\Sigma \subset V$ which inflates the object in the image I and is consistent with its silhouette S. For simplicity, an orthographic projection is assumed and defined by $\pi : V \to \Omega$. In order to handle arbitrary topologies, the surface Σ is represented implicitly by the indicator function $u : V \to \{0, 1\}$ denoting the exterior ($u = 0$) or interior ($u = 1$) of the surface as $u = \mathbf{1}_{int(\Sigma)}$. The semantics and relations of these sets and functions is illustrated in Figure 6.2.

A smooth surface with the desired properties is obtained by minimizing the following energy functional:

$$E(u) = E_{\text{data}}(u) + \nu E_{\text{smooth}}(u) \ , \tag{6.1}$$

where $\nu \geq 0$ is a parameter controlling the smoothness the surface. The smoothness term is imposed via the weighted total variation norm (Definition 2.13)

$$E_{\text{smooth}}(u) = \int_V g(u) \, |\nabla u(\boldsymbol{x})| \, d\boldsymbol{x} \ , \tag{6.2}$$

where the diffusivity $g : V \to \mathbb{R}_{\geq 0}$ can be used to adaptively adjust smoothness properties of the surface in different locations. The range of g needs to be non-negative to maintain the convexity of the model. The data term

$$E_{\text{data}}(u) = \int_V u(\boldsymbol{x}) \ \phi_{\text{shape}}(\boldsymbol{x}) d\boldsymbol{x} + \int_V u(x) \ \phi_{\text{sil}}(\boldsymbol{x}) d\boldsymbol{x} \tag{6.3}$$

realizes two objectives: volume inflation with a shape prior and silhouette consistency.

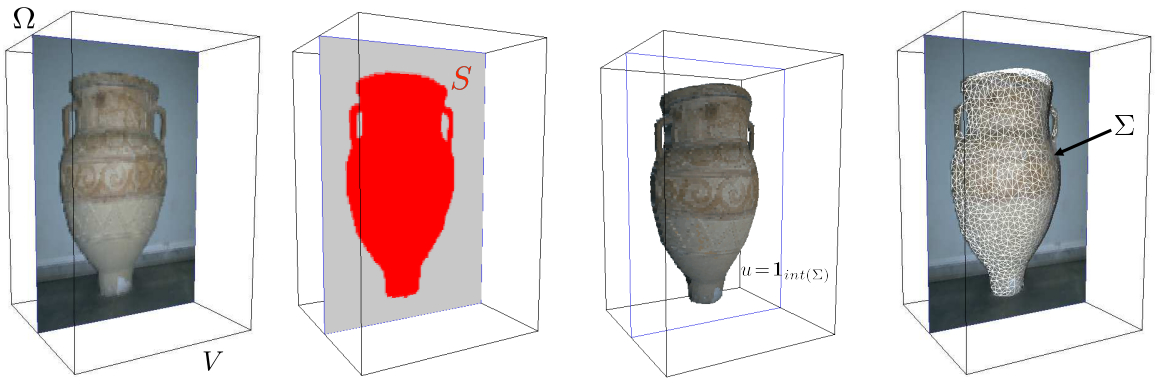


Figure 6.2.: Illustration of our volumetric setup and notation. The image domain Ω is centered within the reconstruction volume V. We constrain the surface Σ, implicitly represented by function u, to be consistent with the silhouette S.

6.2.2. Silhouette Consistency

The function $\phi_{\text{sil}}(\boldsymbol{x})$ merely imposes silhouette consistency. It assures that all points projecting outside the silhouette will be assigned to the background ($u=0$) and that all points which are on the image plane and inside the object will be assigned as object ($u=1$):

$$\phi_{\text{sil}}(\boldsymbol{x}) = \begin{cases} -\infty & \text{if } \boldsymbol{x} \in S \\ +\infty & \text{if } \pi(\boldsymbol{x}) \notin S \\ 0 & \text{otherwise} \end{cases} , \tag{6.4}$$

where $\pi(\boldsymbol{x})$ denotes the orthogonal projection of point $\boldsymbol{x} \in V$ onto the image plane Ω.

6.2.3. Volume Inflation

The volume inflation function ϕ_{shape} allows to impose some guess of the shape of the object. The function can be adopted to achieve any desired object shape and may also be changed by user-interaction later on. In this chapter, we make the simple assumption that the thickness of the observed object increases as we move inward from its silhouette. For any point $\boldsymbol{x} \in V$ let

$$\text{dist}(\boldsymbol{x}, \partial S) = \min_{\boldsymbol{s} \in \partial S} \|\boldsymbol{x} - \boldsymbol{s}\| \quad , \tag{6.5}$$

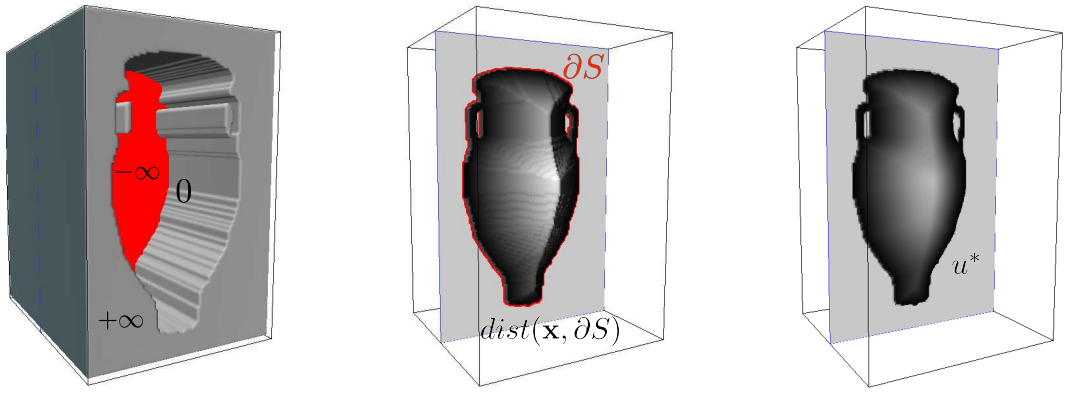
denote its distance to the silhouette contour $\partial S \subset \Omega$. Then we set:

$$\phi_{\text{shape}}(\boldsymbol{x}) = \begin{cases} -1 & \text{if } \text{dist}(\boldsymbol{x}, \Omega) \leq h(\pi(\boldsymbol{x})) \\ +1 & \text{otherwise} \end{cases} , \tag{6.6}$$

where the height map $h : \Omega \to \mathbb{R}_{\geq 0}$ depends on the distance of the projected 3D point to the silhouette according to the function

$$h(\boldsymbol{y}) = \min \left\{ \lambda_{\text{cutoff}}, \lambda_{\text{offset}} + \lambda_{\text{factor}} * \text{dist}(\boldsymbol{y}, \partial S)^k \right\} \tag{6.7}$$

with four parameters $k, \lambda_{\text{offset}}, \lambda_{\text{factor}}, \lambda_{\text{cutoff}} \in \mathbb{R}_{\geq 0}$ affecting the shape of the reconstructed object. How the user can employ these parameters to modify the computed 3D shape will be discussed in Section 6.3.



Silhouette Constraints | Silhouette Contour Distance | Smooth Solution

Figure 6.3.: Illustration of the data term consisting of silhouette constraints and our proposed distance-based shape prior. The surface is forced to be silhouette consistent by setting the gray area (left) to infinity, while the silhouette itself is known to be part of the surface.

Note that this choice of ϕ_{shape} implies symmetry of the resulting model with respect to the image plane. Since the backside of the object is unobservable, it will be reconstructed properly for plane-symmetric objects.

6.2.4. Optimization via Convex Relaxation

To minimize energy (6.1) we follow the framework developed in [135]. To this end, we relax the binary assumption by allowing u to take on intermediate values, i.e. $u : V \rightarrow [0, 1]$. Subsequently, we can globally minimize the convex functional (6.1) by solving the corresponding Euler-Lagrange equation

$$0 = \phi_{\text{shape}} + \phi_{\text{sil}} - \nu \ \text{div}\left(g\frac{\nabla u}{|\nabla u|}\right) \ , \tag{6.8}$$

using the lagged diffusivity fixed-point iteration scheme described in Section 4.3.2. A global optimum of the original binary labeling problem is then obtained by simple thresholding of the solution of the relaxed problem (as described in Chapter 4).

6.3. Interactive Single-View Reconstruction

6.3.1. Interactive Editing

From the input image and silhouette a first reconstruction is generated, which - depending on the complexity and the class of the object - can already be satisfactory. However, for some object classes and due to the general over-smoothing of the resulting mesh, we propose several editing techniques on a 1D (parameter) and a 2D (image space) level. The goal is to have easy-to-use editing tools which cover important cases of object features.

In this chapter we present three different kinds of editing tools: parameter-based, contour-based and curve-based tools. The first two classes operate directly on the data term of Equation (6.1), whereas the third one alters the diffusivity of the TV-norm in Equation (6.2).

Shape Prior Parameters. By altering the parameters $\lambda_{\text{offset}}, \lambda_{\text{factor}}, \lambda_{\text{cutoff}}$ and the exponent k of the shape prior function in Equation (6.6), users can intuitively change the data term in Equation (6.3) and thus the overall shape of the reconstruction. Note that the impact of these parameters is attenuated with increasing importance of the smoothness term. The effects of the *offset*, *factor* and *cutoff* parameters on the shape prior are shown in Figure 6.4 and are quite intuitive to grasp. The exponent k of the distance function in Equation (6.6) mainly influences the objects curvature in the proximity of the silhouette contour. This can be observed in Figure 6.4 showing an evolution from a cone to a cylinder just by decreasing k.

Local Data Term Editing. Due to the use of a distance function for the volume inflation, depth values of the data term will always increase for larger distances to the silhouette contour. Thus, large depth values will never occur near the silhouette contour. However this can become necessary for an important class of object shapes like for instance the bottom and top of the vase in Figure 6.5. A simple remedy to this problem is to ignore user specified contour parts during the calculation of the distance function. We therefore approximate the object contour by a polygon which is laid over the input image. The edges of the polygon are points of high curvature and each edge represents the contour pixels between the endpoints.

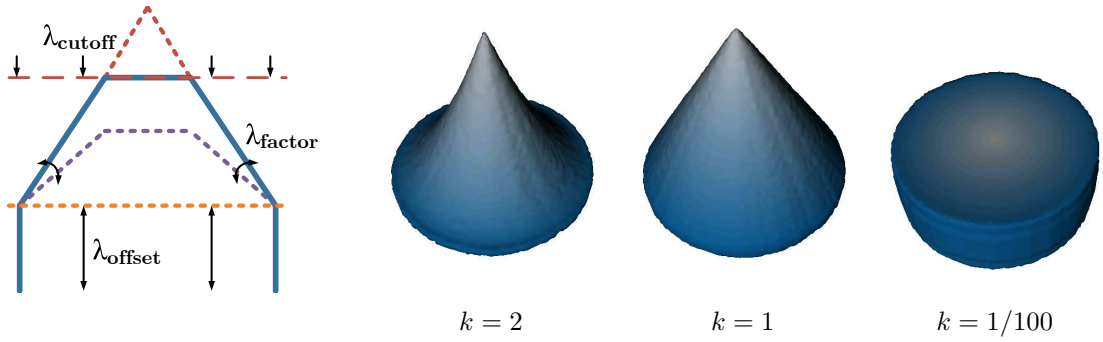


Figure 6.4.: Effect of $\lambda_{\text{offset}}, \lambda_{\text{factor}}, \lambda_{\text{cutoff}}$ (left) and various values of parameter k and resulting (scaled) shape prior plots for a circular silhouette.

Figure 6.5.: Top row: shape priors and corresponding reconstructions with and without marked sharp contour edges. Bottom row: input image with marked contour edges (blue) and line strokes (red) for local discontinuities which are shown right.

By clicking on the edge, the user indicates to ignore the corresponding contour pixels during distance map calculation (see Figure 6.5 top right).

Local Discontinuities. Creases on the surface often add critically to the characteristic shape of an object. With the diffusivity function of the smoothness term in Equation (6.2) we are given a natural way of integrating discontinuities into the surface reconstruction. By setting the values of g to less than one for certain subsets of the domain, the smoothness constraint is relaxed for these regions. Accordingly for values greater than one smoothness is locally fortified. To keep things simple, we let the user specify curves of discontinuities by drawing them directly into the input image space. In the reconstruction space, the corresponding pre-images are uniquely defined hyperplanes (remembering that we make use of parallel projection). For the points lying on these planes or surrounding them, the diffusivity is reduced resulting in a surface crease at the end of the reconstruction process.

6.3.2. Implementation

In order to efficiently solve the Euler-Lagrange Equation (6.8) and allow fast interactive modeling the choice of the solving method and its appropriate implementation is crucial to achieve short calculation times.

Instead of minimizing Equation (6.1) with a gradient descent scheme, we solve the approximated system of linear equations with successive over-relaxation (SOR) as proposed in [135]. On the one hand, this increases the convergence speed drastically and on the other the solution method can be parallelized to further increase computational speed. Therefore, we make use of the CUDA framework to implement SOR with a Red-Black scheme which speeds up calculations by factor 6 compared to the sequential method. Moreover, the computational effort for the surface evolution during interactive modeling can be further reduced by initializing the calculations with the previous reconstruction result. For small parameter changes this initialization is usually close to the next optimal solution. In sum, this allows single-view reconstruction close to realtime.

6.4. Experiments

In the following we apply our method to several input images. We show different aspects of the reconstruction process for typical classes of target objects. Further we mention runtimes and limitations of the approach.

The experimental results are shown in Figure 6.6. Default parameters for the shape prior (Equation (6.6)) are $k = 1$, $\lambda_{\text{offset}} = 0$, $\lambda_{\text{factor}} = 1$, $\lambda_{\text{cutoff}} = \infty$. Each row depicts several views of a single object reconstruction starting with the input image.

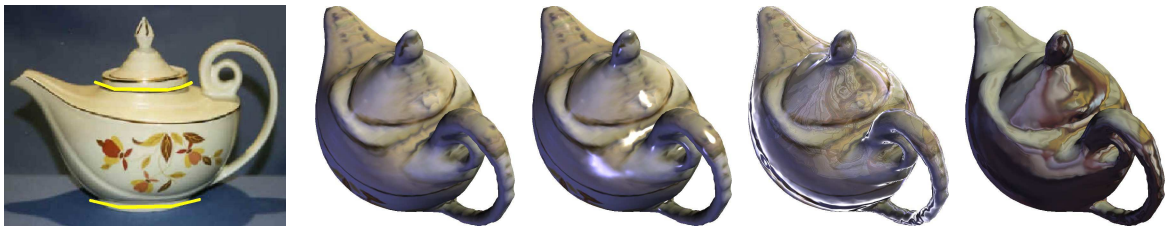
The following main advantages are showcased in the examples. The fence (top row) is an example of an object with complex topology, the algorithm can handle. Obviously reconstructions of the shown type are nearly impossible to achieve with the help of parametrized representations. The same example is also a proof for how little user interaction is necessary in some cases to obtain a good reconstruction result. In fact, the fence was automatically generated by the method right after the user segmentation stage. The rest of the examples demonstrate the power of the editing tools described in Section 6.3. The reconstructions were edited by adding creases and selecting sharp edges. It can be seen, that elaborate modeling effects can be readily achieved with these operations. Especially for the cockatoo a single curve suffices in order to add the characteristic indentation to the beak. No expert knowledge is necessary. For the socket of the Cristo statue, creases help to attain sharp edges, while keeping the rest of the statue smooth. It should be stressed, that no other post-processing operations were used.

The experiments in the lower three rows stand for a more complex series of target objects. A closer look reveals that the algorithm clearly attains its limit. The structure of the opera building (third row) as well as the elaborate geometry of the bike and its drivers cannot be correctly reconstructed with the proposed method due to a lack of information and more sophisticated tools. Yet the results are appealing and could be spiced up with the given tools. To keep the runtime and memory demand within convenient limits, we work on 256^2-input images. These result in a very detailed mesh. On a GeForce GTX card an update step of the geometry takes about 2-15 seconds, dependent on the applied operation.

Figure 6.7 illustrates some possible applications of our single-view reconstruction approach which can be used for image and video editing. Apart from novel view synthesis one can easily change material and reflectance properties as shown in the pictures. Especially relighting of objects is often necessary when they are copied from one image to another. With a plausible

Figure 6.6.: Input images (1st column) and corresponding reconstruction results (2nd-4th column): textured model, untextured geometry, textured model without image plane.

3D reconstruction the relighting is more realistic and probably needs less user-interaction then a manual purely image-based relighting.



Input with User-Strokes | Stone | Porcelain | Varnish | Aluminum

Figure 6.7.: Possible applications of our single-view reconstruction approach. Novel view synthesis and change of material and reflectance properties of the surface.

6.5. Conclusion

In this chapter we presented the first variational approach for single-view reconstruction of curved objects with arbitrary topology. It allows to compute a plausible 3D model for a limited but reasonable class of single images. By using an implicit surface representation we eliminate the dependency on a choice of surface parametrization and the subsequent difficulty with objects of varying topology. The proposed functional integrates silhouette information and additional user input. Globally optimal reconstructions are obtained via convex relaxation. The algorithm can be used interactively, since the parallel implementation of the underlying nonlinear diffusion process on standard graphics cards only requires short runtimes. The minimal surface prior and a plane symmetry prior strongly guide the surface reconstruction process and thus simplify editing and modeling of these aspects. We demonstrated that only few user-defined parameters are necessary to define a shape prior that results in plausible object reconstructions. Compared to other works, the amount of user input is small and intuitive, post-editing is kept simple and does not require expert knowledge.

One disadvantage of the proposed shape prior is the edgy structure of central object parts due to the fact that the applied distance transform has strong discontinuities at points with equal distance to several points of the silhouette boundary. Although these discontinuities are smoothed out locally by the minimal surface prior, they are usually visible on larger scales and could make the user-editing tedious. In the next chapter, we show that perfectly curved objects can be obtained even simpler by exchanging the shape prior with a volume prior which also further decreases the amount of necessary user input.

7. Single-View 3D Reconstruction with a Volume Prior

Simplicity is the ultimate sophistication.

Leonardo da Vinci
(1452-1519)

7.1. Introduction

In the previous Chapter 6 we proposed a shape prior to tackle the inflation problem. Despite a number of convincing results, this work suffers from several drawbacks: Firstly, imposing a thickness proportional to the distance from the silhouette outline is very strong and not always a correct assumption. Secondly, this inflation heuristic has strong discontinuities at points having a similar distance to several points on the silhouette outline. Even for higher smoothness values, this discontinuity is usually apparent in the reconstruction result. Thirdly, the modeling requires a large number of not necessarily intuitive parameters controlling the data term. In this chapter, we show that the shape prior can be replaced with a volume prior which provides an inflation heuristic that is perfectly smooth and requires less tuning parameters. The key idea is to compute a silhouette-consistent weighted minimal surface for a user-specified volume. In this sense, the proposed formulation is closely related to the concept of *Cheeger sets* – sets which minimize the ratio of area over volume [52]. Reconstruction results with discontinuities such as the one Figure 7.1 can still be obtained by additional scribble-based user input in the same manner as in the previous chapter.

Our shape prior approach and the related works on single-view reconstruction mentioned in Section 5.1 have in common that they revert to inflation heuristics in order to avoid surface collapsing. These techniques boil down to fixing absolute depth values, which undesirably restrict the solution space. A precursor to volume constraints are the volume inflation terms pioneered for deformable models by Cohen and Cohen [56]. However, no constant volume constraints were considered and no implicit representations were used.

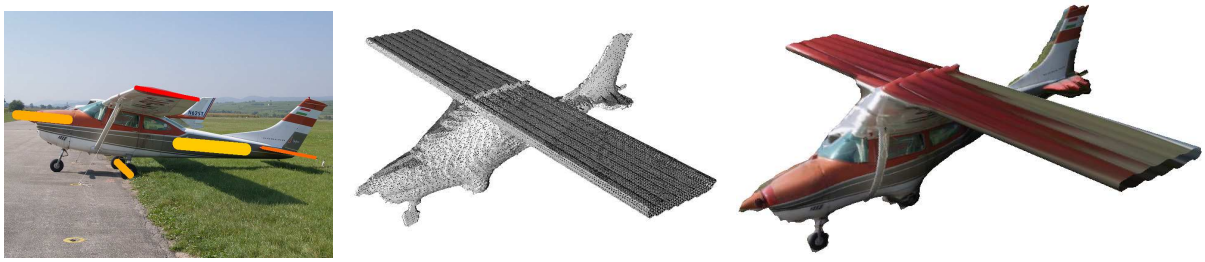


Image with User Input | Reconstructed Geometry | Textured Geometry

Figure 7.1.: The proposed method generates convincing $3D$ models from a single image computed by fixed volume weighted minimal surfaces. Colored lines in the input image mark user input, which locally alters the surface smoothness. Red marks low, yellow marks high smoothness (see Section 7.4.4 for details).

Contributions. The main contributions of this chapter can be summarized as follows:

- We propose a weighted minimal surface approach for single-view reconstruction with a volume constraint for surface inflation. To the best of our knowledge this is the first work on convex shape optimization with guaranteed volume preservation.
- We show that the fixed-volume minimal surface problem is a convex problem which solutions can be shown to be within provable energetic bounds from the optimal solution.
- Due to the volume constraint no further inflation heuristics are required. The amount of user input is significantly reduced.

The work in this chapter was published in [3] and was part of a comparison to the previous shape prior approach and other related works in [4, 6].

7.2. Fixed-Volume Minimal Surface Formulation

In this section, we will drop the data term that was used for surface inflation in the previous Chapter 6. Therefore, the silhouette consistency of the surface will be enforced by means of constraints to the minimal surface problem.

The weighted minimal surface problem is posed by minimizing the total variation over a suitable set U of feasible indicator functions $u \in \mathcal{BV}(V, \{0,1\})$:

$$u^* = \arg\min_{u \in U} \int_V g(\boldsymbol{x}) |\nabla u(\boldsymbol{x})| \, d\boldsymbol{x} \ , \tag{7.1}$$

where ∇u denotes the derivative in the distributional sense and the surface smoothness is locally affected by the weighting function $g(\boldsymbol{x}) : V \to \mathbb{R}_{\geq 0}$ which can be used for further optional user modeling.

How does the set U of feasible functions look like? For simplicity, we assume the silhouette to be enclosed by the surface. Then all surface functions that are consistent with the silhouette S must be in the set

$$U_S = \left\{ u \in \mathcal{BV}(V, \{0,1\}) \;\middle|\; u(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \pi_\Omega(\boldsymbol{x}) \notin S \\ 1 & \text{if } \boldsymbol{x} \in S \\ 0 \text{ or } 1 & \text{otherwise} \end{cases} \right\} \tag{7.2}$$

Minimizing Equation (7.1) with respect to the set U_S of silhouette consistent functions will result in the silhouette itself. In the following section we will show a way to avoid this trivial solution.

7.2.1. Volume Constraint

In order to inflate the solution of Equation (7.1) we propose to use a constraint on the size of the volume enclosed by the minimal surface. We formulate this both as a soft- and as a hard constraint and discuss the two approaches in the following.

Hard Constraint. By further constraining the feasible set U_S one can force the reconstructed surface to have a specific target volume V_t. We regard the problem

$$u^* = \underset{u \in U_S \cap U_V}{\arg\min}\, E(u) \qquad \text{with} \quad E(u) = \int_V g(\boldsymbol{x}) |\nabla u(\boldsymbol{x})|\, d\boldsymbol{x} \tag{7.3}$$

$$\text{and} \quad U_V = \left\{ u \in \mathcal{BV}(V, \{0,1\}) \;\middle|\; \int_V u(\boldsymbol{x})\, d\boldsymbol{x} = V_t \right\} \tag{7.4}$$

where U_V denominates all reconstructions with bounded variation that have the specific volume V_t.

Soft Constraint. For the sake of completeness we also consider the soft formulation of the volume constraint. One can add a ballooning term to Equation (7.1):

$$E_V(u) = \lambda \left(\int_V u(\boldsymbol{x})\, d\boldsymbol{x} - V_t \right)^2 \tag{7.5}$$

The integral quadratically punishes the deviation of the surface volume from a certain target volume V_t. In contrast to the constant volume constraint above, this formulation comes with an extra parameter λ which is why in the following we will focus on the hard constraints in Equation (7.3) instead.

Different approaches to finding V_t can be considered. In the implementation the optimization domain is naturally bounded. We choose V_t to be a fraction of the volume of this domain. In a fast interactive framework the user can then adapt the target volume with the help of instant visual feedback. Most importantly, as opposed to a data term driven model volume constraints do not dictate where inflation takes place. Intuitively, this approach to single-view reconstruction corresponds to a balloon being placed inside the silhouette-constrained domain and being inflated to a given volume.

7.2.2. Fast Minimization

In order to convexify the problem in Equation (7.3) we make use of the relaxation technique in [47], which is explained in Section 4.1. To this end we relax the binary range of functions u in Equation (7.2) and Equation (7.4) to the interval $[0, 1]$. In other words we replace U_V and U_S with their respective convex hulls U_V^{rel} and U_S^{rel}. The corresponding optimization problem is then convex:

Proposition 7.1. *The relaxed set $U^{\text{rel}} := U_S^{\text{rel}} \cap U_V^{\text{rel}}$ is convex.*

Proof. The constraint in the definition of U_V is clearly linear in u and therefore U_V^{rel} is convex. The same argument holds for U_S. Being an intersection of two convex sets U^{rel} is convex as well. □

One standard way of finding the globally optimal solution to this problem is gradient descent, which is known to converge very slowly. Since optimization speed is an integral part of an interactive reconstruction framework, we employ a recently proposed significantly faster and provably convergent primal-dual algorithm published in [173]. The scheme is based on the

weak formulation of the total variation:

$$u^*_{\text{rel}} = \min_{u \in U^{\text{rel}}} \int_V g(\boldsymbol{x}) |\nabla u| \, d\boldsymbol{x} = \min_{u \in U^{\text{rel}}} \sup_{|\boldsymbol{p}(\boldsymbol{x})|_2 \leq g(\boldsymbol{x})} \left\{ \int_V -u \operatorname{div} \boldsymbol{p} \; d\boldsymbol{x} \right\} \tag{7.6}$$

Optimization is done by alternating a gradient descent with respect to the function u and a gradient ascent for the dual variable $\boldsymbol{p} \in \mathcal{C}^1_c(\mathbb{R}^3, \mathbb{R}^3)$ interlaced with an over-relaxation step on the primal variable:

$$\begin{cases} \boldsymbol{p}^{k+1} = \Pi_{|\boldsymbol{p}(\boldsymbol{x})|_2 \leq g(\boldsymbol{x})}(\boldsymbol{p}^k + \tau \cdot \nabla \bar{u}^k) \\ u^{k+1} = \Pi_{U^{\text{rel}}}(u^k + \sigma \cdot \operatorname{div} \boldsymbol{p}^{k+1}) \\ \bar{u}^{k+1} = 2u^{k+1} - u^k \end{cases} \tag{7.7}$$

where Π_A denotes the projection onto the set A. Projection of $\boldsymbol{p}$ is done by simple clipping while that of the primal variable u will be detailed in the next paragraph. The scheme in Equation (7.7) is numerically attractive since it avoids division by the potentially zero-valued gradient-norm which appears in the Euler-Lagrange equation of the TV-norm. Moreover, it is parallelizable and we therefore implemented it on the GPU. On a volume of 63x47x60 voxels the computation takes only 0.47 seconds.

7.2.2.1. Projection Scheme

The projection $\Pi_{U^{\text{rel}}}$ in Equation (7.7) needs to ensure three constraints on u: Silhouette consistency, constant volume and $u \in [0, 1]$. In order to maintain silhouette consistency (Equation (7.2)) of the solution we restrict updates to those voxels which project onto the silhouette interior excluding the silhouette itself.

For the projection $\Pi_{U^{\text{rel}}}(u)$ we make use of the algorithm by Boyle and Dykstra [31] which computes the Euclidean projection of a point onto the intersection of arbitrary convex sets and provably converges to the projection solution.

Formally, for our case step i of this algorithm reduces to two separate projections for volume and unit domain

$$\begin{cases} u^i_V = u^{i-1}_R - v^{i-1}_V + \frac{V_d}{|V|} \\ v^i_V = u^i_V - (u^{i-1}_R - v^{i-1}_V) \end{cases} \tag{7.8}$$

$$\begin{cases} u^i_R = \Pi_{[0,1]}(u^i_V - v^{i-1}_R) \\ v^i_R = u^i_R - (u^i_V - v^{i-1}_R) \end{cases}, \tag{7.9}$$

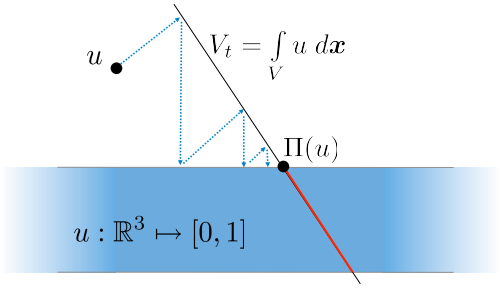


Figure 7.2.: Illustration of the projection scheme by Boyle and Dykstra [31]. Alternating projection onto different convex sets finally leads to the projection onto their intersection.

where we initialize u_R with the current u^k in Equation (7.7) and v_R, v_V with zero. $\Pi_{[0,1]}(u)$ simply clips the value of u to the unit interval and $V_d := \int_V u \, d\boldsymbol{x} - V_t$ is the difference between the target volume V_t and the current volume of the values $u^{i-1}_R - v^{i-1}_V$. $|V|$ is the number of voxels in the discrete implementation. In Equation (7.9) u^i_R represents the current estimate which converges towards $\Pi_{U^{\text{rel}}}(u)$ with increasing iterations i.

7.2.3. Optimality Bounds

Having computed a global optimal solution u^*_{rel} of the relaxed problem in Equation (7.6), the question remains how we obtain a binary solution and how the two solutions relate to one another energetically. Unfortunately no thresholding theorem holds, which would imply energetic equivalence of the relaxed optimum and its thresholded version for arbitrary thresholds.

Nevertheless we can construct a binary solution u_{thr} via thresholding, but the threshold no has to fulfill the volume constraint and can be found as follows:

Proposition 7.2. *The relaxed solution can be projected to the set of binary functions in such a way that the resulting binary function preserves the user-specified volume V_t.*

Proof. It suffices to order the voxels $\boldsymbol{x}$ by decreasing values $u(\boldsymbol{x})$. Subsequently, one sets the value of the first V_t voxels to 1 and the value of the remaining voxels to 0. □

Concerning an optimality bound the following holds:

Proposition 7.3. *Let u^*_{rel} be the global optimal solution of the relaxed energy and u^*_{bin} the global optimal solution of the binary problem. Then*

$$E(u_{\text{thr}}) - E(u^*_{\text{bin}}) \leq E(u_{\text{thr}}) - E(u^*_{\text{rel}}) \ . \tag{7.10}$$

A proof and a discussion of this relation was given in Section 4.1.

7.3. Theoretical Analysis of Material Concentration

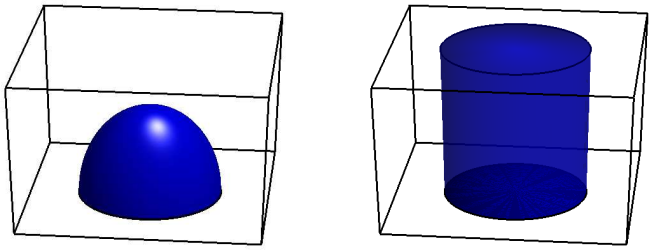


Figure 7.3.: The two cases considered in the analysis of the material concentration. On the *left* hand side we assume a hemi-spherical condensation of the material. On the *right* hand side the material is distributed evenly over the volume.

As we have seen above, the proposed convex relaxation technique does not guarantee global optimality of the binary solution. The thresholding theorem [47] – applicable in the unconstrained problem – no longer applies to the volume-constrained problem. While the relaxation naturally gives rise to a-posteriori optimality bounds, one may take a closer look at the given problem and ask why the relaxed volume labeling u should favor the emergence of solid objects rather than distribute the prescribed volume equally over all voxels.

In the following, we will prove analytically that the proposed functional has an energetic preference for material concentration. For simplicity, we will consider the case that the object silhouette in the image is a disk. And we will compare the two extreme cases of all volume being concentrated in a ball (a known solution of the Cheeger problem) compared to the case that the same volume is distributed equally over the feasible space (namely a cylinder) – see Figure 7.3.

Input Image | Reconstruction | +25% Volume | +35% Volume

Figure 7.4.: The inflation of the reconstruction model can be intuitively changed by varying the target volume.

Proposition 7.4. *Let u_{sphere} denote the binary solution which is 1 inside the sphere and 0 outside – Figure 7.3, left – and let u_{cyl} denote the solution which is uniformly distributed (i.e. constant) over the entire cylinder – Figure 7.3, right. Then we have*

$$E(u_{sphere}) < E(u_{cyl}), \tag{7.11}$$

independent of the height of the cylinder.

Proof. Let r denote the radius of the disk. Then the energy of u_{sphere} is simply given by the area of the half-sphere:

$$E(u_{sphere}) = \int_V |\nabla u_{sphere}| \, d\boldsymbol{x} = 2\pi r^2. \tag{7.12}$$

If instead of concentrated to the half-sphere, the same volume, i.e. $v = \frac{2\pi}{3} r^3$, is distributed uniformly over the cylinder of height $h \in (0, \infty)$, we have

$$u_{cyl}(x) = \frac{v}{\pi r^2 h} = \frac{2\pi r^3}{3\pi r^2 h} = \frac{2}{3}\frac{r}{h}. \tag{7.13}$$

inside the entire cylinder, and $u_{cyl}(x) = 0$ outside the cylinder. The respective surface energy of u_{cyl} is given by the area of the cylinder weighted by the respective jump size:

$$E(u_{cyl}) = \int_V |\nabla u_{cyl}| \, d\boldsymbol{x} = \left(1 - \frac{2r}{3h}\right) \pi r^2 \; + \; \frac{2r}{3h}(\pi r^2 + 2\pi r h) = \frac{7}{3}\pi r^2 > E(u_{sphere}). \tag{7.14}$$

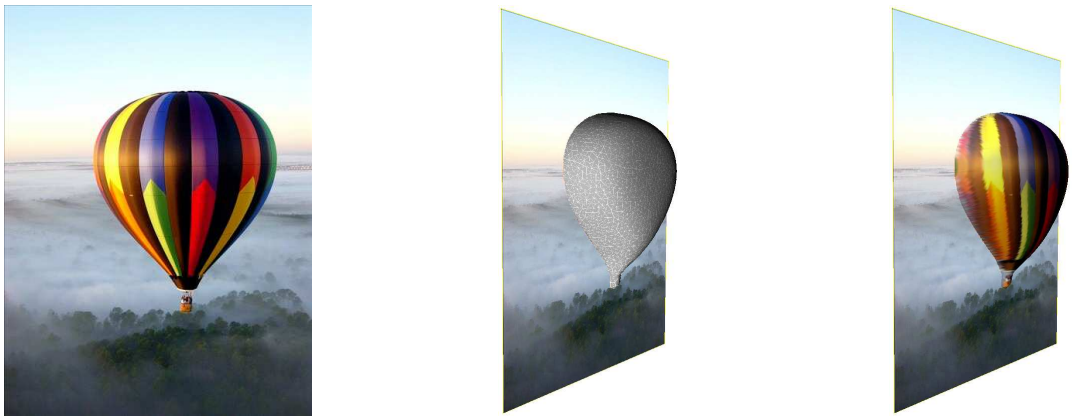
□

7.4. Experimental Results

Having detailed the idea of variational implicit weighted surfaces and their fast computation, in this section we will study their properties and applicability within an interactive reconstruction environment. We will compare our approach to methods which resort to heuristic inflation techniques and finally show that appealing and realistic 3D models can be generated with minimal user input.

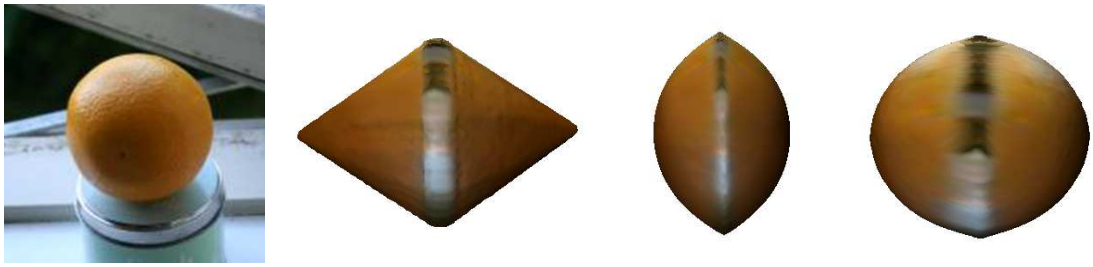
7.4.1. Cheeger Sets and Single-View Reconstruction

Solutions to Equation (7.3) are Cheeger sets [52], that is, minimal surfaces for a fixed volume. In the simplest case of a circle-shaped silhouette one therefore expects to get a ball. Figure 7.5 demonstrates that in fact round silhouette boundaries (in the unweighted case) result in round shapes.



Input Image | Reconstructed Geometry | Textured Geometry

Figure 7.5.: The proposed volume prior approach favors minimal surfaces for a user-specified volume. Therefore the reconstruction algorithm is ideally suited to compute smooth, round reconstructions.



Input Image | Data Term as Shape Prior | Reconstruction with Data Term | Reconstruction with Volume Prior

Figure 7.6.: Using a silhouette distance transform as shape prior the relation between data term (*second from left*) and reconstruction (*third from left*) is not easy to assess for a user. With only one parameter our method delivers more intuitive and natural results.

7.4.2. Fixed Volume vs. Shape Prior

Many approaches to volume reconstruction incorporate a shape prior in order to avoid surface collapsing. A common heuristic is to use a distance transform of the silhouette boundary for depth value estimation. We show that the fixed-volume approach solves several problems of such a heuristic.

Figure 7.6 shows that it is hard to obtain ball-like surfaces with a silhouette distance transform as a shape prior. Another issue is the strong bias a shape prior inflicts on the reconstruction resulting in cone-like shapes (see Figure 7.7) and inhibiting the flexibility of the model. The uniform fixed-volume approach fills both gaps while exhibiting the favorable properties of the distance transform (as seen in Figure 7.9). With the results in Figure 7.6 and Figure 7.7 we directly compare our method to [1] and [183], in which the reconstruction volume is inflated artificially.

7.4.3. Varying the Volume

Apart from the weighting function of the TV-norm (see next section), the only parameter we have to determine for our reconstruction is the target volume V_t. The effect on the appearance of the surface can be witnessed in Figure 7.4. One can see that changing the target volume has an intuitive effect on the resulting shape which is important for a user driven reconstruction.

Input Image | Reconstruction with Shape Prior | Reconstruction with Volume Prior

Figure 7.7.: In contrast to the shape prior approach [1], the proposed volume prior does not favor a specific shape and generates more pleasing $3D$ models. Although in the center reconstruction the dominating shape prior can be mitigated by a higher smoothness, this ultimately leads to the vanishing of thin structures like the handle.

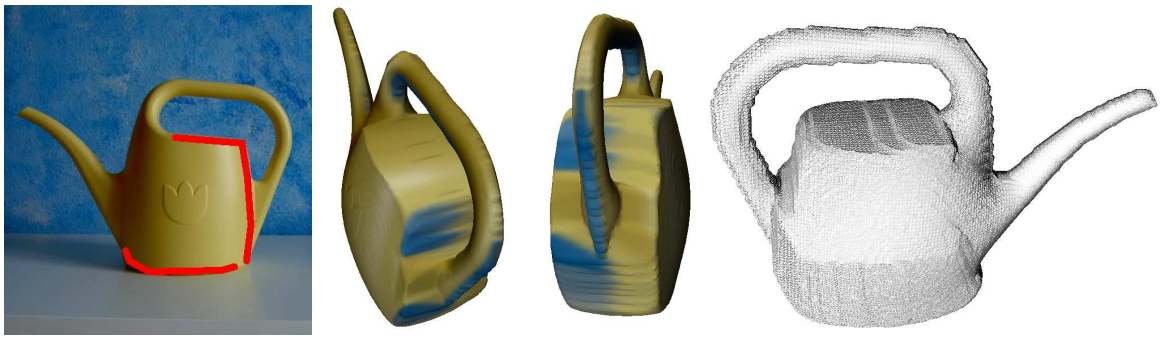


Image with User Input | Reconstructions | Geometry

Figure 7.8.: The proposed volume prior also allows to generate $3D$ models with sharp edges. The red user strokes define locations for which the surface can be non-smooth by down-weighting the corresponding costs in the regularizer (see Section 7.4.4 for more details).

7.4.4. Weighted Minimal Surface Reconstruction

So far all presented reconstructions came along without further user input. The weight $g(x)$ of the TV-norm in Equation (7.6) can be used to locally control the smoothness of the reconstruction: with a low $g(x)$, the smoothness condition on the surface is locally relaxed, allowing for creases and sharp edges to form. Conversely, setting $g(x)$ to a high value locally enforces surface smoothness. For controlling the weighting function we employ a user scribble interface. The parameter associated to each scribble marks the local smoothness within the respective scribble area and is propagated through the volume along projection direction. In Figure 7.8 we show that with this tool not only round, but other very characteristic shapes can be modeled with minimal user interaction.

The air plane in Figure 7.1 represents an example, where a parametric shape prior would fail to offer the necessary flexibility required for modeling protrusions. Since our fixed-volume approach does not impose points of inflation, user input can influence the reconstruction result in well-defined ways: Marking the wings as highly non-smooth (i.e. low $g(x)$) effectively allows them to form. Note that apart from Figures 7.1 and 7.8 the adaption of the target volume was the only user input for all experiments.

In Figure 7.10 we compare the proposed volume prior with the shape prior from the previous Chapter 6 on a variety of objects. The results look mostly similar, but the different inflation heuristics are often distinguishable, because the shape prior via distance transform has a discontinuity at points that have similar distance to several boundary points. As a result, the reconstructions with the volume prior exhibit smoother surfaces. A more thorough comparison of the two approaches and other state-of-the-art methods will be given later in Chapter 9.

Input Reconstruction Different View Geometry

Figure 7.9.: Volume inflation dominates where the silhouette area is large (bird) whereas thin structures (twigs) are inflated less.

7.5. Conclusion

In this chapter, we presented a novel framework for single-view reconstruction which allows to compute $3D$ models from a single image in form of Cheeger sets, i.e. minimal surfaces for a fixed user-specified volume. The framework allows for appealing and realistic reconstructions of curved surfaces with minimal user input. The reconstruction problem is posed as finding a silhouette-consistent minimal surface with a user-specified volume. The resulting convex energy is optimized globally using an efficient provably convergent primal-dual scheme. A parallel GPU implementation allows for computation times of a few seconds, allowing the user to interactively increase or decrease the volume. We proved that the computed surfaces are within a bound of the optimum and that they exactly fulfill the target volume. On a variety of challenging real world images, we showed that the proposed method compares favorably over existing implicit approaches, that volume variations lead to families of realistic reconstructions and that additional user scribbles allow to locally reduce smoothness so as to easily create protrusions.

The proposed approach also has several drawbacks. Similar to the shape prior approach (Chapter 6) the volumetric surface representation needs considerable memory resources and is computationally expensive due to the high number of variables to be optimized. Therefore, the image and depth resolution is limited which may lead to noticeable discretization artifacts. Another disadvantage is that the volume-preserving thresholding scheme does not guarantee optimality of the original binary problem, that is, the thresholding Theorem 4.1 does not hold in combination with the volume constraint. In the next chapter, we will show that the same energy with a parametric surface representation tackles all these disadvantages.

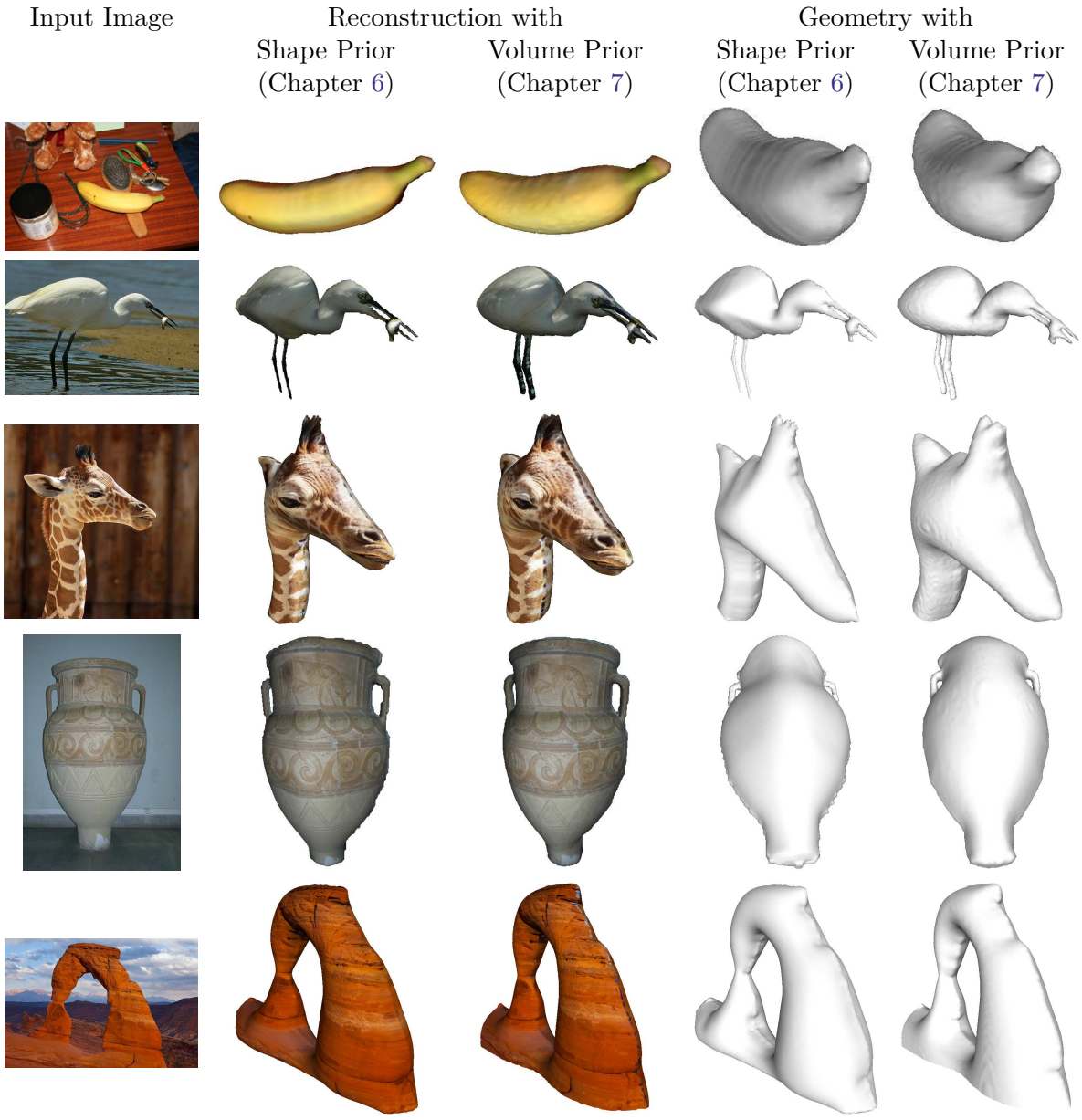


Figure 7.10.: Output of single-view methods with the shape prior (Chapter 6) in comparison with the volume prior (Chapter 7) for several examples.

8. Single-View 2.5D Reconstruction with a Volume Prior

Chi conosce la geometria, può compremendere tutto in questo mondo.

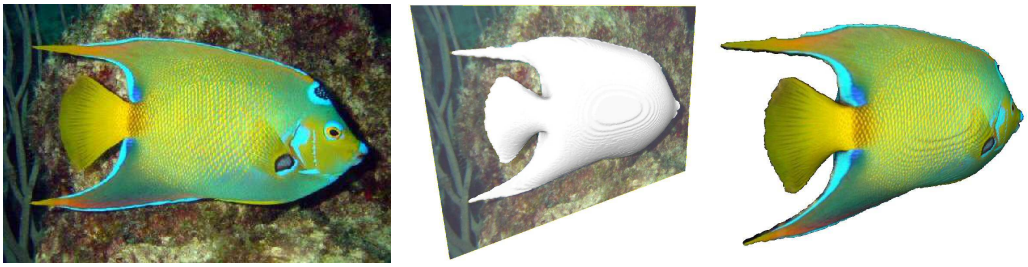
Galileo Galilei
(1564-1642)

8.1. Introduction

In this chapter, we show that the single-view method with the volume prior from the previous Chapter 7 can be solved more efficiently and accurately by using a parametric 2.5D surface representation, rather than an implicit 3D surface representation. In comparison to the implicit method, the optimization becomes globally optimal and reconstruction results, such as the one in Figure 8.1, can be obtained by an order of magnitude faster. With a 2.5D representation, we refer to a height map that is parametrized in 2D and assigns a height in the third dimension to every point in the domain.

In short, the 3D implicit approach from the previous Chapter 7 has the following drawbacks:

- The volumetric representation imposes strong constraints on memory and runtime. Even with an efficient GPU-accelerated primal-dual algorithm the method requires around a second of computation time for moderate resolution reconstructions. As a consequence, higher-resolution 3D models cannot be generated at interactive runtimes.
- Although the method in the previous Chapter 7 was shown to provide exactly volume-consistent solutions, the algorithm is based on convex relaxation and thresholding. In the absence of a threshold theorem, the method is not guaranteed to provide the globally minimal surface of specified volume. Furthermore, it is not clear whether subsequent thresholding of the relaxed solution actually leads to a spatially coherent structure (rather than a scattered set of voxels).
- The method is essentially computing a height map, because the advantages of a fully volumetric representation are not used. The required discretization of possible depth values imposes a limitation on the possible resolution in the z-direction.



Input Image | Reconstructed Geometry | Textured Geometry

Figure 8.1.: The proposed algorithm computes optimal silhouette-consistent minimal surfaces of given volume in computation times below 1s.

Contributions. In this chapter, we propose a novel algorithm for computing single-view reconstructions which remedies the above shortcomings. More precisely:

- We propose to solve the above problem by means of a height-field representation. As a consequence, we can allow for a spatially continuous set of depth values.
- Due to the $2.5D$ surface representation we have substantially reduced computation time and memory requirement (quadratic rather than cubic). Experiments confirm that the proposed method allows to compute solutions about an order of magnitude faster, even for higher resolutions.
- In contrast to the fully volumetric approach in Chapter 7, the proposed method does not require convex relaxation and thresholding. As a consequence, the algorithm provably computes silhouette-consistent minimal surfaces of a specified volume.

The method in this chapter was published in [5].

8.2. Fixed Volume Minimal Surfaces on a Two-Dimensional Grid

The main difference to the fully volumetric approach is the surface representation. The objects' surface will be represented by means of a height map

$$u : S \subset \Omega \to \mathbb{R}_{\geq 0} \tag{8.1}$$

assigning a height value $u(x, y)$ to each point $(x, y) \in S$ of the silhouette which is embedded in the image domain $\Omega \subset \mathbb{R}^2$.

As shown in the schematic plot in Figure 8.2, an infinitesimal surface area element ds of the surface represented by the function u can be computed as the area of the parallelogram via the cross product and is given by

$$ds = \left| \begin{pmatrix} dx \\ 0 \\ u_x\, dx \end{pmatrix} \times \begin{pmatrix} 0 \\ dy \\ u_y\, dy \end{pmatrix} \right| = \sqrt{1 + |\nabla u|^2}\, dx\, dy \tag{8.2}$$

where u_x and u_y are abbreviations for the corresponding partial derivatives of the height map, that is $\nabla u = (u_x, u_y)^T = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)^T$. The overall area of the surface denoted by u is given by

$$E(u) = \int_\Sigma ds = \int_S \sqrt{1 + |\nabla u|^2}\, dx\, dy \tag{8.3}$$

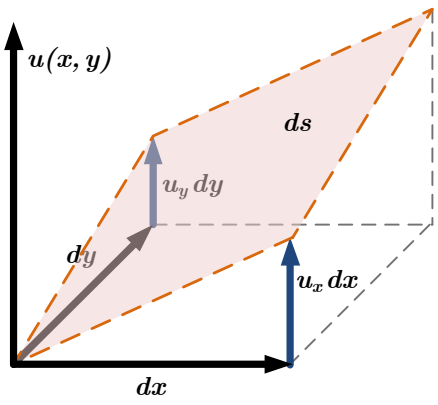


Figure 8.2.: The area of an infinitesimal surface element ds based on the partial derivatives of u.

This minimal surface energy has first been derived by Joseph Louis Lagrange [147, p.354ff] in the year 1760, but without any additional constraints. More studies of this energy can be found in [118, p.8] and [65].

Again, for brevity $d\boldsymbol{x}$ will denote a vectorial integrand being two-dimensional in this chapter. Reconstructing a minimal surface enclosing a given target volume V_t can therefore be

expressed as the minimization problem

$$u^* = \arg\min_{u \in U} E(u), \quad \text{with} \quad U = \left\{ u \in \mathcal{C}^1(S, \mathbb{R}_{\geq 0}) \,\middle|\, \int_S u \, d\boldsymbol{x} = V_t \right\}. \tag{8.4}$$

Note, that the target volume V_t in the volume constraint effectively defines the object's *average* depth value, since the volume is the product of the average depth value and the silhouette area.

Proposition 8.1. *The two-dimensional fixed volume minimal surface problem defined in Equation (8.4) is convex.*

Proof. The volume constraint in Equation (8.4) is linear in u, and thus defines a convex optimization domain. Moreover, the functional E in Equation (8.4) is convex. This can be shown by using the triangle inequality, the linearity of the gradient operator and the zero-order convexity condition in Definition 2.3. For any functions u_1 and u_2 and any $\alpha \in (0, 1)$ the following inequality holds

$$\begin{aligned}
&E(\alpha u_1 + (1-\alpha)u_2) \\
&= \int \sqrt{1 + \left|\nabla(\alpha u_1 + (1-\alpha)u_2)\right|^2} \, d\boldsymbol{x} \\
&= \int \sqrt{1 + \left|\alpha \nabla u_1 + (1-\alpha)\nabla u_2\right|^2} \, d\boldsymbol{x} \\
&= \int \left\| \begin{pmatrix} \alpha \nabla u_1 + (1-\alpha)\nabla u_2 \\ \alpha + (1-\alpha) \end{pmatrix} \right\| d\boldsymbol{x} \\
&= \int \left\| \alpha \begin{pmatrix} \nabla u_1 \\ 1 \end{pmatrix} + (1-\alpha) \begin{pmatrix} \nabla u_2 \\ 1 \end{pmatrix} \right\| d\boldsymbol{x} \\
&\leq \int \alpha \left\| \begin{pmatrix} \nabla u_1 \\ 1 \end{pmatrix} \right\| + (1-\alpha) \left\| \begin{pmatrix} \nabla u_2 \\ 1 \end{pmatrix} \right\| d\boldsymbol{x} \\
&= \int \alpha \sqrt{1 + |\nabla u_1|^2} + (1-\alpha)\sqrt{1 + |\nabla u_2|^2} \, dx \\
&= \alpha E(u_1) + (1-\alpha) E(u_2) \ .
\end{aligned} \tag{8.5}$$

□

In contrast to the volumetric formulation proposed in [3], the two-dimensional formulation proposed here is convex. As a consequence, we do not need to revert to the generally suboptimal strategy of convex relaxation and thresholding. Instead we can directly compute globally optimal solutions by solving (8.4).

8.3. Minimization of the Proposed Energy

Minimization of the convex problem in Equation (8.4) can be achieved by solving the Euler-Lagrange extremality condition given by the partial differential equation

$$\frac{dE}{du} = -\operatorname{div}\left(\frac{\nabla u}{\sqrt{1 + |\nabla u|^2}} \right) = 0. \tag{8.6}$$

This is a nonlinear diffusion equation which is similar to the well-known model by Perona and Malik [170] for edge-preserving image smoothing, but with a different diffusivity $g(x) =$

$1/\sqrt{1+|\nabla u|^2}$ which was proposed by Charbonnier et al. [51].

Our derivation of Equation (8.6) via Equation (8.3) therefore provides a geometric interpretation of the Perona-Malik diffusion with the Charbonnier-diffusivity: In image diffusion the image gray values can be interpreted as a height map whose surface area is minimized as the diffusion process minimizes energy (8.3) (see also [200] for more details).

However, we use Equation (8.6) in a completely different setting. Instead of using a data term we impose a global volume constraint and special boundary conditions which depend on the input silhouette. In the following we describe how these constraints are chosen and incorporated into the numerical optimization of Equation (8.6).

8.3.1. Numerical Optimization

We employed three optimization schemes and compared their performance. We briefly sketch all three methods in the following. In [59] Paul Concus proposed a numerical scheme for solving the minimal surface problem (8.4) except that he did not consider a volume constraint.

Projected Gradient Descent. As explained in Section 4.3.1, gradient descent is the simplest numerical solver for differentiable objective functions. Since our domain is restricted by the linear volume constraint, we have to apply projected gradient descent in order to stay within the feasible domain. In each iteration we advance in the direction of the negative gradient of the energy, and back-project onto the feasible set U step:

$$u^{k+1} = \Pi_U \left[u^k - \tau \frac{dE(u^k)}{du} \right] \tag{8.7}$$

where τ is the step size and $\Pi_U(u)$ is the Euclidean projection of u onto the set U being defined in the next subsection. Since the minimization problem (8.4) is convex, the gradient descent method will converge to the global optimum of the energy.

Fast Iterated Shrinkage and Thresholding Algorithm (FISTA). This algorithm by Beck and Teboulle [20] (see Section 4.3.3) can be considered as a generalized gradient descent scheme for a certain class of functions. Applied to our case it amounts to an ordinary projected gradient descent with an adaptive over-relaxation step:

$$\begin{aligned} u^k &= \Pi_U \left[\bar{u}^k - \frac{1}{L} \frac{dE(\bar{u}^k)}{du} \right] \\ \tau^{k+1} &= \frac{1}{2} \left(1 + \sqrt{1 + 4(\tau^k)^2} \right) \\ \bar{u}^{k+1} &= u^k + \left(\frac{\tau^k - 1}{\tau^{k+1}} \right) (u^k - u^{k-1}) \ . \end{aligned} \tag{8.8}$$

The parameter L is the Lipschitz constant of the functional E and defines the step width of the descent scheme.

Lagged Diffusivity Fixed Point Iteration (LDFPI). The Lagged-diffusivity approach by Vogel and Oman [232] (see Section 4.3.2) solves the Euler-Lagrange equation like a Quasi-Newton method. By keeping the diffusivity $g(\boldsymbol{x}) = \sqrt{1+|\nabla u|^2}$ in Equation (8.6) fix over a number of iterations one can solve the resulting sparse linear equation system $\operatorname{div}(g(\boldsymbol{x})\nabla u) = 0$ with numerical solvers like Jacobi, Gauss-Seidel or Successive Over-Relaxation (SOR).

We update the diffusivity every few iterations and project the solution onto the feasible set U. Note, that due to the projection this scheme will not provably converge to the optimal solution, but, as we will show later, it turned out to be the fastest solver which leads to visually similar results.

8.3.2. Implementation

In order to solve the optimization problem in Equation (8.4) with one of the three optimization methods from above we proceed as follows: We compute one or more iterations of our optimization algorithm and then project the current solution back to the convex set U of functions with a pre-described volume.

Projection Scheme. The orthogonal projection of any function u onto U can be described as the following optimization problem:

$$\Pi_U(u) = \arg\min_{u'} \frac{1}{2} \int_S \|u - u'\|^2 \, dx \quad \text{s.t.} \quad \int_S u \, d\boldsymbol{x} = V_t \ . \tag{8.9}$$

By introducing the Lagrange multiplier $\lambda \in \mathbb{R}$ and calculating the partial derivatives of the corresponding Lagrangian function we obtain the following extremality conditions:

$$0 = u - u' + \lambda \quad \forall \boldsymbol{x} \in S \tag{8.10}$$

$$0 = \int_S u \, d\boldsymbol{x} - V_t \tag{8.11}$$

Inserting Equation (8.10) into Equation (8.11) yields

$$\Pi_U(u) = u + \Big(\frac{V_t - \int_S u \, d\boldsymbol{x}}{\int_S d\boldsymbol{x}} \Big) \cdot \mathbf{1}_S \tag{8.12}$$

as a simple update scheme for the volume projection. Function $\mathbf{1}_S$ is again the indicator function (Definition 3.2) which is 1 at every point $\boldsymbol{x} \in S$ and 0 otherwise. Equation (8.12) means that the residual volume is evenly distributed over all function values of u in S.

Boundary Conditions. In order to guarantee silhouette consistency induced by the set S, we apply Dirichlet boundary conditions at the silhouette boundary ∂S and Neumann boundary conditions if the silhouette coincides with the image boundary $\partial\Omega$:

$$u \Big|_{\partial S \setminus \partial\Omega} = 0 \qquad \text{and} \qquad \nabla_{\boldsymbol{n}} u \Big|_{\partial\Omega} = \langle \nabla u, \boldsymbol{n} \rangle \Big|_{\partial\Omega} = 0 \ , \tag{8.13}$$

where $\nabla_{\boldsymbol{n}}$ is the directional derivative in orthogonal direction $\boldsymbol{n} \in \mathbb{R}^2$ to the image boundary. This way silhouette consistency is ensured and objects touching the image boundary are cut orthogonal to the image plane. Intuitively, this means that object surfaces continue uniformly at image boundaries rather than dropping to zero.

Parallelization. All minimization methods described in Section 8.3.1 have been parallelized on recent graphics hardware. This includes the projection step since it can be applied to each pixel independently once the difference between target and current volume is known. For parallelization of the LDFPI method with SOR a Red-Black scheme has been employed.

8.3.3. Weighted Minimal Surfaces

Without adding further constraints to the solution, the problem in (8.4) tends to be smooth by definition. In order to enable our method to reconstruct non-smooth objects, we can add local weights to the energy functional. More formally Equation (8.3) extends to

$$E(u) = \int_S \rho(\boldsymbol{x})\sqrt{1 + |\nabla u|^2}\, d\boldsymbol{x} \ . \tag{8.14}$$

Fortunately, the introduction of the weighting function $\rho : S \to \mathbb{R}_{\geq 0}$ does not affect the convexity of the energy.

Proposition 8.2. *The two-dimensional fixed volume minimal surface problem defined in Equation (8.4) extended with the weighting function as shown in Equation (8.14) is convex.*

Proof. The proof is a straight-forward extension of the one from Proposition 8.1. □

Further, this extension is easily integrated into the optimization methods described above. Adding weights to the surface considerably extents the class of possible reconstructions. Setting all weights $\rho(\boldsymbol{x}) = 1$ leads to the original formulation in Equation (8.3). In the implementation we use this as a default setting, however, the user can locally adapt this surface parameter.

8.4. Experimental Results

We tested our method on several real-world images, compared the results with our full 3D single-view reconstruction approach from the previous Chapter 7, and two other state-of-the-art methods. Further, we evaluated visual appearance, runtime and amount of user input.

Since one cannot obtain true depth values from a single image we do not strive for a comparison with ground truth data. We rather focus on plausibility and pleasantness of the reconstructions. Again, we assume the reconstructions to be symmetric with respect to the image plane in order to reconstruct the backsides of objects as well. Due to the symmetry assumption, we are also able to obtain closed object representations from height maps.

8.4.1. Qualitative Comparison to Related Methods

In Figure 8.4 we visually compare our results to the ones obtained with the methods by Zhang et al. [253], Prasad et al. [183] and our 3D single-view approach with a volume prior (Chapter 7/[3]). For comparison we used the implementation from [253]. We do not have an implementation of Prasad et al. [183] and therefore used the results presented in [181].

The method by Zhang et al. [253] sticks out in this comparison because it is restricted to depth map reconstructions while the other methods focus on curved 3D objects. Except for our 3D single-view method [3] all approaches are globally optimal and compute reconstructions at interactive frame rates. The methods mostly differ in the necessary amount of user input.

For the method of Zhang et al. [253], the user has a variety of choices for surface manipulations such as position and normal constraints, discontinuity constraints, planar region constraints and manual mesh-subdivision. Usually many of these constraints are necessary for reasonable reconstructions leading to modeling times of several minutes to hours even for experienced users. We, as moderately experienced users, spent 20-40 minutes for each of the examples shown in Figure 8.4.

Similarly, the method by Prasad et al. [183] needs concise input and expert knowledge. The user has to assign parts of contour lines to lines in the parameter space, which becomes harder for objects of higher genus. As a result, the topology is restricted to genus two. Still, objects of higher genus exhibit over-oscillation of the surface as seen in the teapot example in Figure 8.4. Moreover, for volume inflation the user needs to define a set of interpolation constraints. In subsequent steps the user may need to add further constraints for allowing surface creases. On the other hand and in contrast to our approach, Prasad et al. [183] can cope better with some images, in which the symmetry plane of the object is not parallel to the image plane. An example for this is represented by the donut in Figure 8.4.

Our single-view method with volume prior on the full 3D volume being described in the previous Chapter 7 minimizes a similar energy and need the same amount of user input, which is considerably less compared to the other reconstruction methods. Several examples in Figures 8.4 and 8.5 compare both approaches. Since the 2.5D method needs less memory and computation time, it is feasible to use input images with considerably higher resolution. This results in higher detailed silhouettes and reconstructions as can be seen in the plane example in Figure 8.7. Also, results of our method appear smoother as we compute continuous depth values (see e.g. the balloon). In contrast, the full 3D single-view method [3] scales poorly with the input image size since a voxel field needs more memory and runtime resources.

8.4.2. Experimental Evaluation of our Approach

Figures 8.4 and 8.5 show reconstruction results of our method for various input images. The examples represent objects of very different quality reaching from natural to man-made objects. One can see that the reconstructions appear quite plausible.

In general, since we compute a minimal surface, reconstructions will often exhibit a balloonish appearance. However, the final minimal surface strongly depends on the shape of the input silhouette. With regard to this, a strength of our approach is that volume is inflated naturally in correspondence to silhouette compactness. Examples for this favorable behavior are the bird, the stone arch and also the teapot in Figure 8.4. They show that parts of the silhouette that are compact inflate more, whereas thinner structures are inflated less.

All the examples in Figure 8.5 and Figure 8.4 come without smoothness adaption (see Section 8.3.3). In these cases, the only parameter of our approach is the volume of the reconstruction. Figure 8.6 visualizes how changes of the target volume V_t intuitively affect the shape of the reconstruction.

In the other cases the user changed the smoothness of the surface locally. User scribbles define the locations for which the weighting factor $\rho(\boldsymbol{x})$ of Equation (8.14) can be set to a user defined value. Setting $\rho(\boldsymbol{x})$ to less than 1 locally allows for sharp edges and surface extrusions like the airplane wings in Figure 8.7, while values larger than 1 have the opposite effect of creating indentations. We employed this mechanism as part of an interactive feature in our single-view reconstruction tool. Remember that $\rho(\boldsymbol{x}) = 1$ everywhere the user did not specify weighted regions.

Figure 8.7 shows some results for which the user altered the smoothness locally. One can see that non-smooth reconstructions can be achieved intuitively. Next to each reconstruction the corresponding user scribbles are shown.

Runtime Comparison. As described in Section 8.3, we employed a gradient descent scheme, FISTA and LDFPI for solving problem (8.4). All experiments have been done on a PC with a 2.27GHz Intel Xeon CPU, 12GB RAM equipped with a NVidia GeForce GTX480 graphics card running a recent Linux distribution. For comparing runtimes of the respective

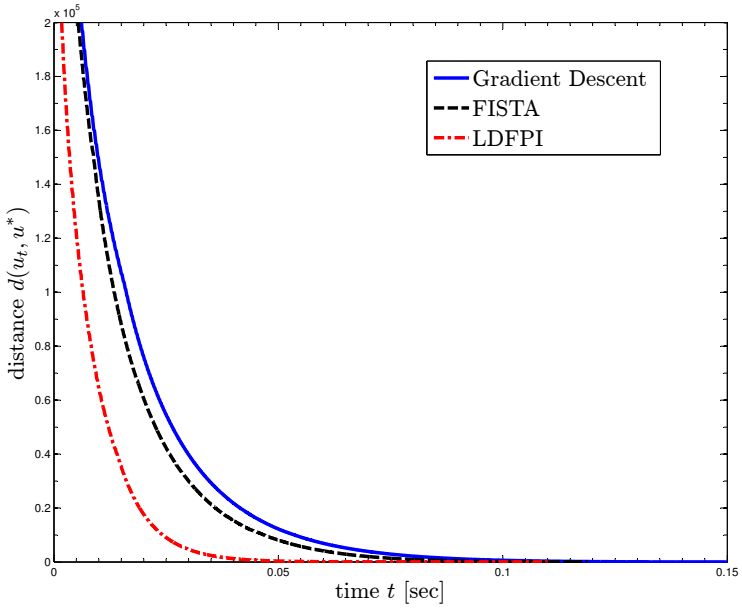


Figure 8.3.: Runtime comparison of different algorithms minimizing Equation (8.4) measured on the teapot example without user-scribbles.

example		Volume Prior 3D [3]	Volume Prior 2.5D [5]	speedup
teapot	size	131x101x58	131x101	
	time	1.82s	0.14s	13.0
arch	size	179x137x79	179x137	
	time	6.24s	0.99s	6.3
ladybug	size	151x122x27	151x122	
	time	1.62s	0.15s	10.8
bird	size	157x244x4	157x244	
	time	2.12s	0.2s	10.6
balloon	size	82x97x44	82x97	
	time	2.65	0.15s	17.7

Table 8.1.: Runtime comparison of the 3D approach [3] with the 2.5D approach with volume prior [5] for the examples depicted in Figures 8.5 and 8.6.

optimization algorithms, we ran each on a reconstruction example until convergence. We then plotted for each time step t the distance $d(u_t, u^*)$ of the intermediate result u_t to the precomputed converged result u^*.

$$d(u_t, u^*) := \int_\Omega \left(u_t(\boldsymbol{x}) - u^*(\boldsymbol{x})\right)^2 d\boldsymbol{x} \tag{8.15}$$

The convergence criterion for all experiments has been set to

$$\left| \frac{E(u_{t-1}) - E(u_t)}{E(u_t)} \right| < \theta \ , \tag{8.16}$$

with $\theta = 10^{-15}$. Figure 8.3 shows the results for the three optimization schemes. As can be clearly seen, the LDFPI approach of Section 8.3 is the most efficient algorithm in terms of time to convergence. The FISTA algorithm is only slightly faster than gradient descent. This is due to the fact that for differentiable functionals the algorithm degrades to a gradient

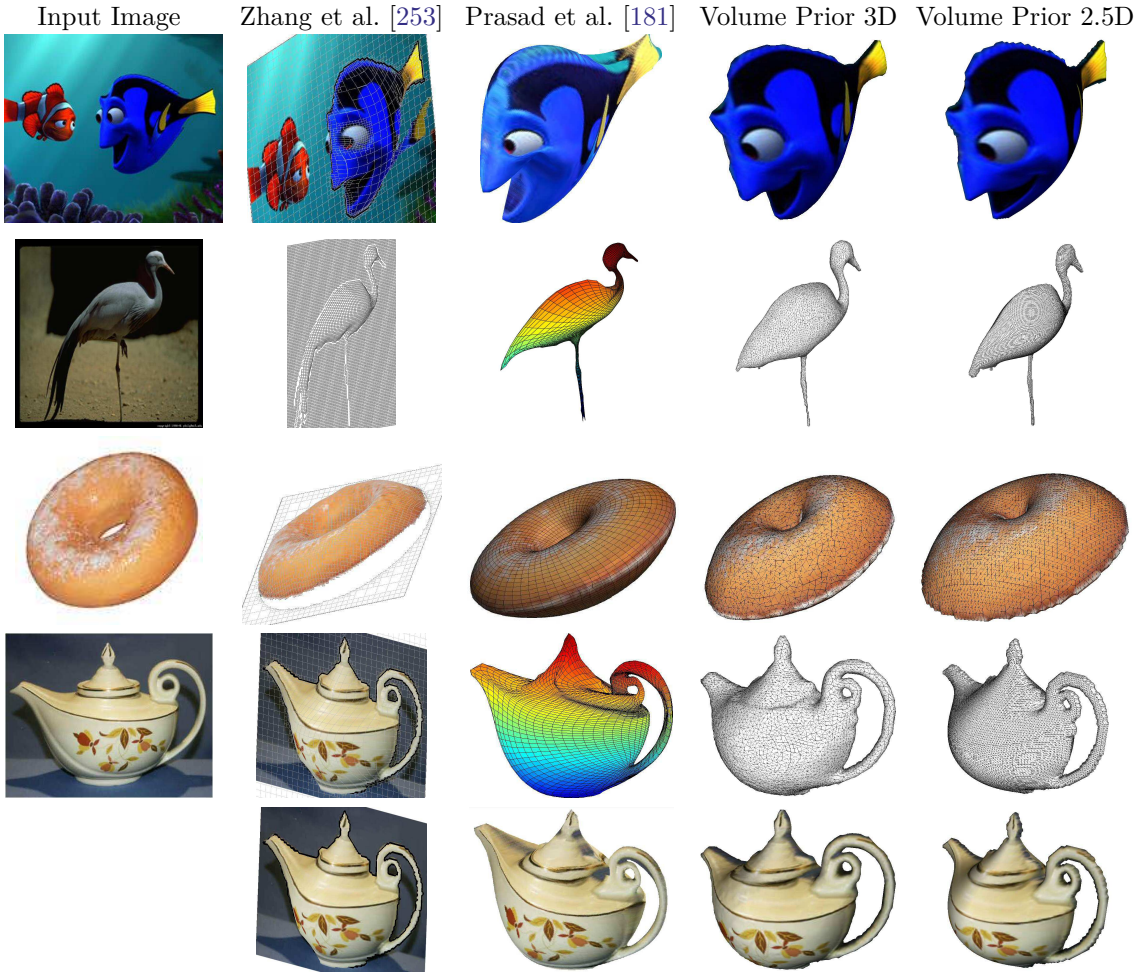


Figure 8.4.: Comparison of reconstruction results for several single-view methods. Qualitatively our methods (*right column*) keeps up with state-of-the-art methods and sometimes even compares favorable over them.

descent. The performance gain stems from the adaptive over-relaxation step. Note that due to the constraints on the feasible set, we have no proof that LDFPI converges to the global optimum (see *Projection Scheme*). However, the results of LDFPI were almost equal to results from methods attaining the global optimum.

In order to evaluate the overall computational efficiency of our method we measured the computation times of the fastest optimization scheme until convergence and compared them with the 3D volume prior approach (Chapter 7/[3]). Table 8.1 shows detailed runtime comparisons for all experiments in Figure 8.5. Since both methods optimize a convex energy, the results are independent of the initialization. The number of iterations needed until convergence, however, is not.

For all experiments the empty surface, respectively the empty volume, has been used for initialization. When the user changes the target volume on a computed result, we can initialize the re-computation cycle with the previously computed solution. This will effectively result in a faster convergence. For input silhouettes with large areas, like the stone arch, the diffusion process has to propagate along longer distances, which leads to the relatively high runtime.

Generally, Table 8.1 clearly shows that the 2.5D approach is significantly faster than the 3D approach. This difference mainly stems from the additional dimension that is used in the latter in order to discretize the depth values while 2.5D approach directly computes continuous solutions.

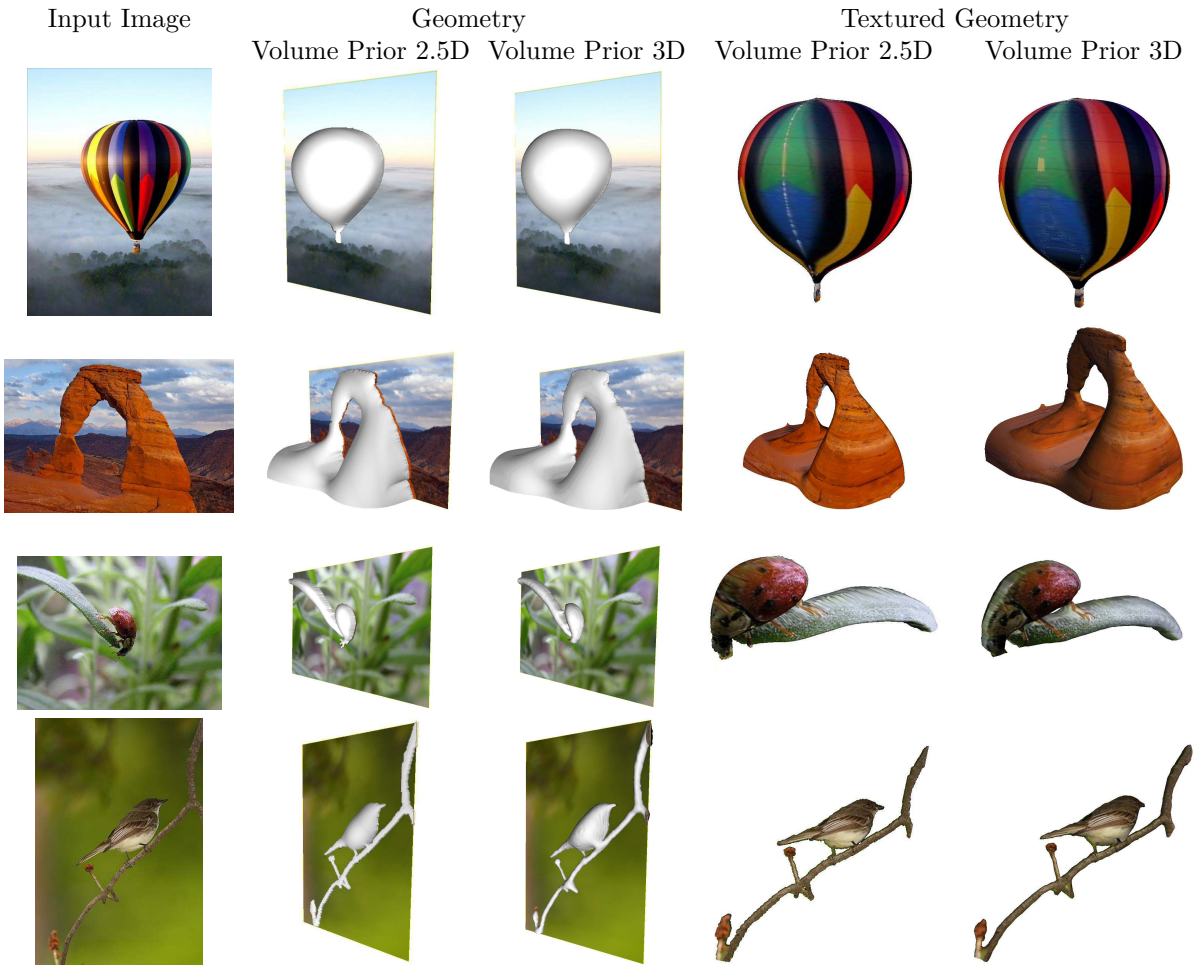


Figure 8.5.: Reconstruction results of the proposed 2.5D approach are similar to our full 3D approach [3] from the previous Chapter 7 but in contrast are obtained for higher resolutions, less memory, lower computation times and higher precision.

Figure 8.6.: Influence of the volume parameter on the reconstruction. The volume distributes naturally, with more volume on compact silhouette parts and less on thin silhouette structures. The input image for this reconstruction is the arch depicted in Figure 8.5

8.5. Conclusion

In this chapter we showed that the 3D single-view approach with a volume prior from the previous Chapter 7 can equivalently be computed by means of a 2.5D height map as surface model. In contrast to the implicit 3D approach from Chapter 7, the proposed 2.5D approach has three advantages: First, the resolution in the depth dimension no longer needs to be explicitly discretized as depth values are directly computed. Secondly, the computed solution is provably optimal (rather than suboptimal). Thirdly, the 2.5D formulation drastically reduces memory and computation time by about an order of magnitude. For a large variety

Figure 8.7.: Reconstruction results with user input altering the local smoothness of the surface. Next to the reconstructions the input images are shown with the respective user scribbles. User scribbles (yellow) decrease the surface smoothness locally.

of objects and good image resolutions, plausible reconstructions are computed in fractions of a second, making this method well suited for interactive $3D$ modeling from images.

In the following chapter, we will compare our single-view reconstruction approaches to other state-of-the-art methods theoretically and experimentally.

9. Comparison of Approaches

Beauty in things exists merely in the mind which contemplates them.

David Hume
(Philosopher, 1711 - 1776)

9.1. Comparison of Approaches for Curved Surface Reconstruction

In this chapter we focus on selected methods from Section 5.1 that aim for the reconstruction of curved surfaces and compare them theoretically and experimentally. In particular, we discuss the methods by Zhang et al. [253], Prasad et al. [183], Igarashi et al. [117] and the ones proposed in this thesis (Chapters 6 to 8). This chapter is part of the publication in [6].

9.1.1. Theoretical Comparison

In the following we will compare the aforementioned approaches with respect to four topics which are important in surface reconstruction.

The Inflation Problem. A common problem of approaches for curved surface reconstruction is that reconstructions tend to be flat since - by default - there are no inflation forces present due to a lack of depth information. A remedy is to let the user specify the depth of certain constraint points of the reconstruction, which are then interpolated by the minimal surface [253, 183], Chapter 6/[1]. This is tedious for the user. The depth constraints can be estimated fully automatically from the silhouette only for cylindrical objects as is done in some examples by Prasad et al. [181]. Several heuristics are conceived for more complicated cases. Igarashi et al. [117] automatically set the depth by a heuristic based on a signed distance function of the silhouette outline. A similar heuristic is used in Chapter 6/[1] in order to define a data term for the variational minimal surface approach. However, in contrast to Igarashi et al. [117] the user is able to adapt the parameters of this data term and thus the final surface. Our proposed volume prior in Chapters 7 and 8 leads in many cases to natural inflation behavior.

Surface Representation and Topology. The reconstructability of curved surfaces with arbitrary topology depends on the surface representation. Our proposed implicit surface representation in Chapters 6 and 7 is generally better suited for this task than parametric ones [183, 253], since the parameter space has to reflect the topology. The same holds for mesh-based approaches such as the one by Igarashi et al. [117]: during modeling operations it can be tedious to keep the mesh consistent, especially during topology changes. The parametric representation by Prasad et al. [183] has other implications. Firstly, uniformly distributed points in the parameter space are not uniformly distributed on the surface. This property may lead to oscillations, especially in the case of higher genus. Further, the relation between points in parameter space and points on the surface is non-trivial for inexperienced users.

Silhouettes. Similar to our single-view reconstruction approaches (Chapters 6 to 8) Prasad et al. [183] used silhouettes for surface inference. Full silhouette consistency of the reconstruction, however, is only enforced in our approaches, because Prasad et al. [183] derive merely local constraints from the silhouette.

View Constraints. Finally, view constraints are of practical importance. All our single-view approaches assume plane symmetric objects. Reconstructions work best if symmetry and viewing plane are parallel. This implies that the contour generator is planar. The approach by Prasad et al. [183] allows for non-planar contour generators and, thus, in some cases for slightly more general view points than just a side-view.

9.1.2. Experimental Comparison

In this section we experimentally compare the methods discussed in the previous subsection. For all experiments, we used the single-view modeling tool by Zhang et al. [253] and the software called SmoothTeddy which incorporates results of several works by Igarashi et al.: [117, 115, 116] which are both publicly available. The reconstruction results by Prasad et al. are taken from the works [182, 183, 181]. In the experiments we only compare our proposed shape prior (Chapter 6) against the volume prior (Chapter 7). The height map approach with a volume prior (Chapter 8) uses a different surface representation as the implicit approach (Chapter 7), but effectively minimizes the same energy and produces very similar results.

In Figures 9.1 to 9.3 we compare the reconstruction results of the five methods on ten different examples, covering various types of objects and related method issues such as object shape, topology, viewing angle and image type. Instead of explaining every example, we rather concentrate on the examples which demonstrate properties, advantages or drawbacks discussed in the theoretical comparison as well as issues we identified during the experiments.

User Input and Modeling Time. Since the necessary amount of user input and thus the simplicity of the modeling process is of particular interest for practical purposes, we also explain and compare the user input for each method.

The method by **Zhang et al.** [253] is more a single-view modeling tool rather than a reconstruction tool. Every detail, every extrusion or inflation has to be modeled by the user. Figure 9.4 illustrates the variety of different constraints listed in Table 9.2 and their general effects on the surface shape. None of these constraints is required by the method, but for most reconstructions a reasonable amount of constraints will be necessary. The large amount of user input results in higher modeling times which are shown in Table 9.1. The difficulty of modeling a non-side view considerably increased the modeling time for the donut example (Figure 9.2). This tool needs user experience.

In contrast, the modeling with the method by **Igarashi et al.** [117] is very simple and fast. None of the user input in Table 9.2 needs much experience or even expert knowledge: From a given closed contour line the method instantly inflates a 3D mesh. See Figure 9.4 for the exemplary user input of the teapot example. In all experiments this method needed the least user input (cf. Table 9.1) at the price of producing the least accurate reconstructions with respect to the given input silhouette (see Figures 9.1 to 9.3).

The user input for the method by **Prasad et al.** [183] is versatile. Most of the user input listed in Table 9.2 is illustrated in Figure 9.5. After the *contour extraction* (Figure 9.5 (a)) normal constraints are generated along the contour shown as red discs with needles in Figure 9.5 (d),(g),(h). The definition of corresponding parameter space boundaries and the assignment of contour line parts to lines in the parameter space is shown in Figure 9.5 (b),(c).

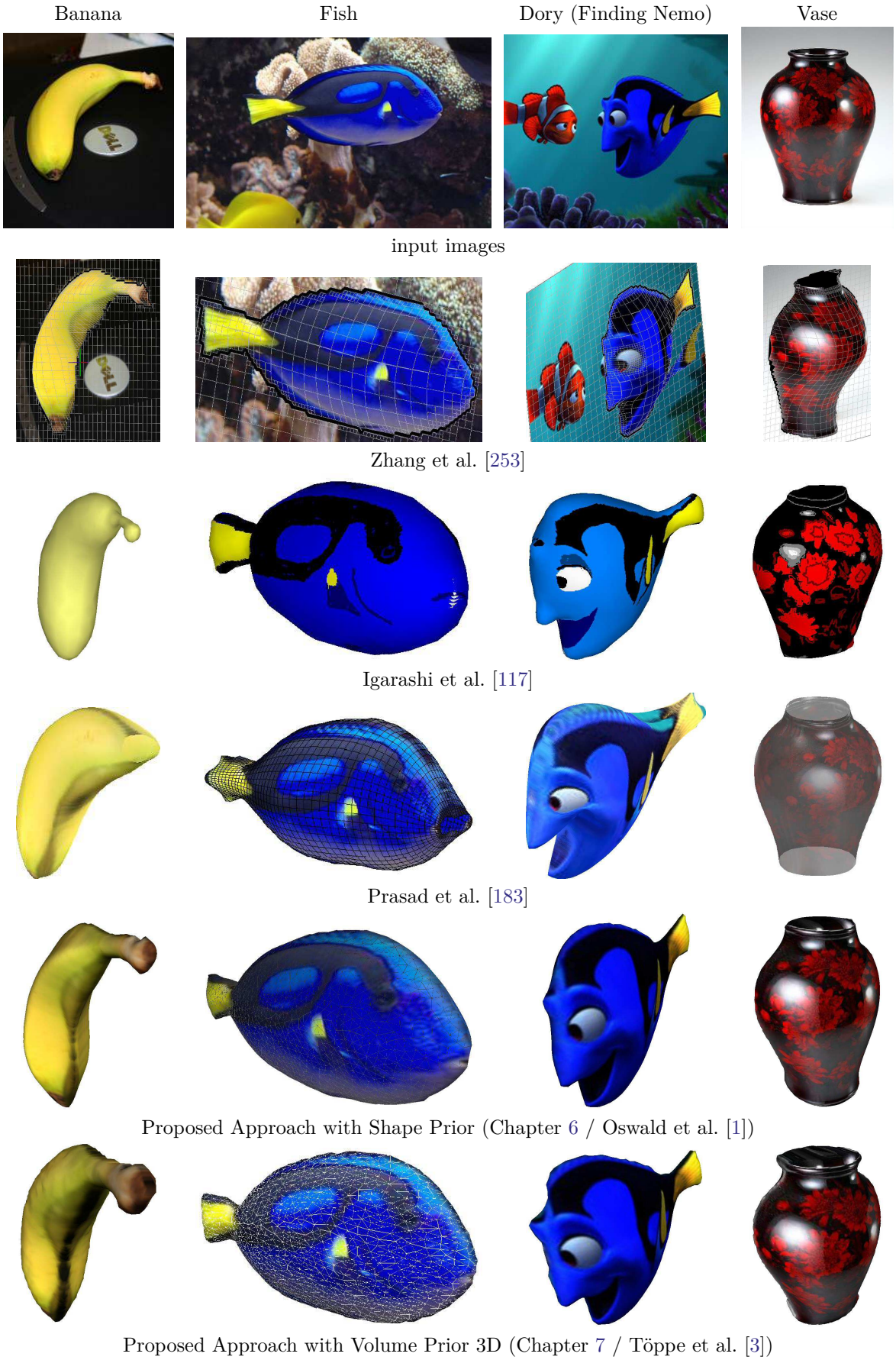


Figure 9.1.: Experimental comparison of several methods for curved object reconstruction. The Figures for Prasad et al. are taken from [181].

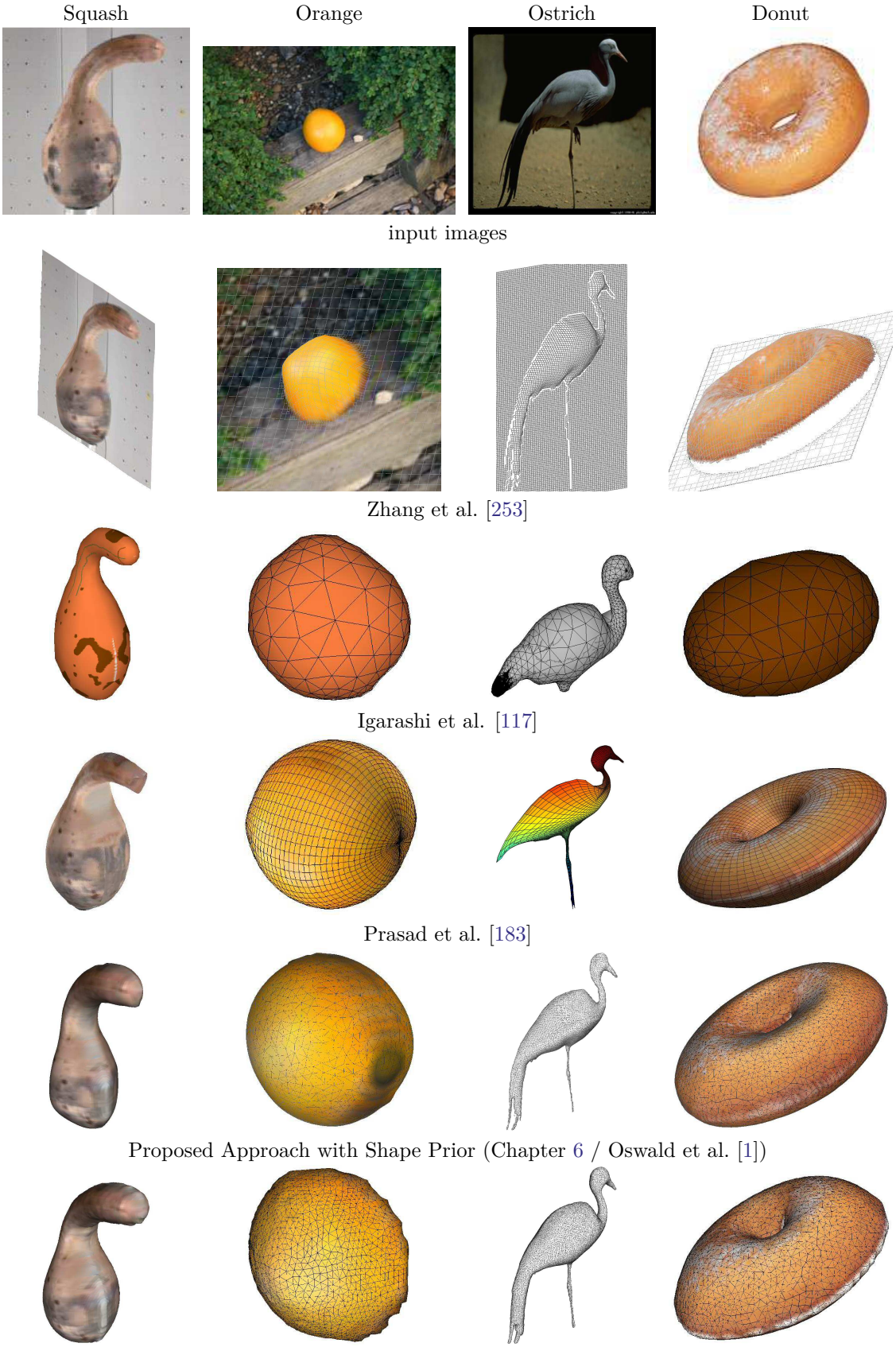


Figure 9.2.: Continuation of Figure 9.1: Experimental comparison of several methods for curved object reconstruction. The Figures for Prasad et al. are taken from [181].

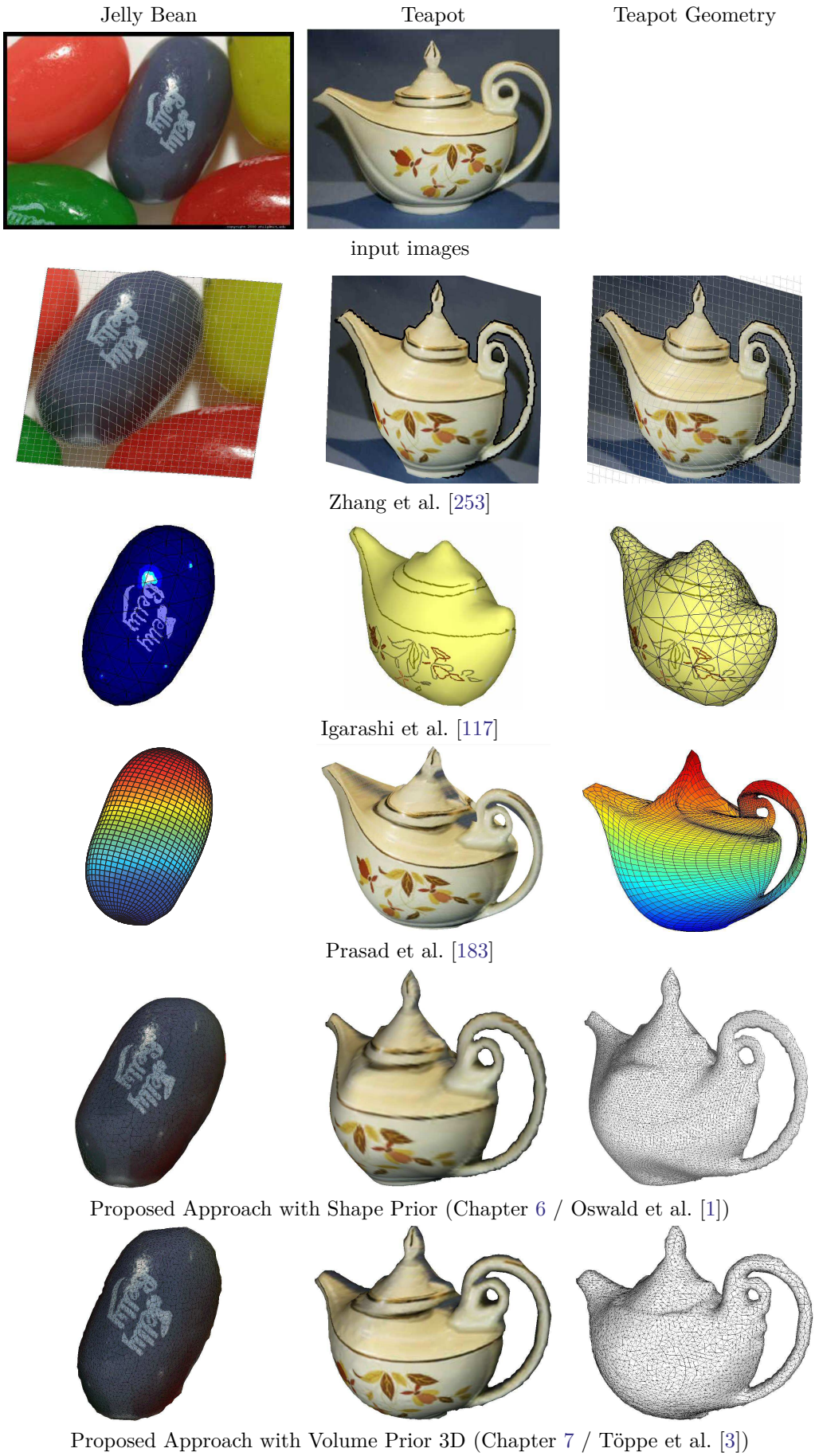


Figure 9.3.: Continuation of Figure 9.2: Experimental comparison of several methods for curved object reconstruction. The Figures for Prasad et al. are taken from [181].

Example	Zhang et al. [253]	Igarashi et al. [117]	Prasad et al. [183]	Shape Prior [1]	Volume Prior 3D [3]	Volume Prior 2.5D [5]
Banana	20 min	<1 min	10 min	5 min	<1 min	<1 min
Fish	15 min	<1 min	2 min	8 min	1 min	<1 min
Dory	40 min	<1 min	5 min	7 min	1 min	1 min
Vase	20 min	<1 min	2 min	13 min	4 min	3 min
Squash	12 min	<1 min	2 min	2 min	1 min	1 min
Orange	14 min	<1 min	<1 min	3 min	<1 min	<1 min
Ostrich	30 min	<1 min	15 min	7 min	2 min	2 min
Donut	55 min	<1 min	10 min	3 min	1 min	1 min
Jelly Bean	15 min	<1 min	2 min	4 min	1 min	1 min
Teapot	35 min	<1 min	20 min	15 min	4 min	3 min

Table 9.1.: Approximate modeling times for a medium experienced user for the examples shown in Figures 9.1 to 9.3. Together with these reconstruction results this table reveals significant differences in the efficiency of the methods on the presented examples.

Objects with cylindrical shape about a virtual free-form 3D curve or spine can be inflated by generating interpolation constraints along the spine with a depth value equal to the minimal distance between the point of the spine and the contour line. This inflation heuristic is generalized for more complex objects as object parts can be independently inflated with the same technique. To this end the user defines pairs of *inflation curves* (Figure 9.5 (e)) for which interpolation constraints are generated along the (virtual) medial spine (Figure 9.5 (g),(h)). The necessity and complexity of each single user input step depends on the object to be reconstructed leading to very different modeling times for the presented experiments (see Table 9.1).

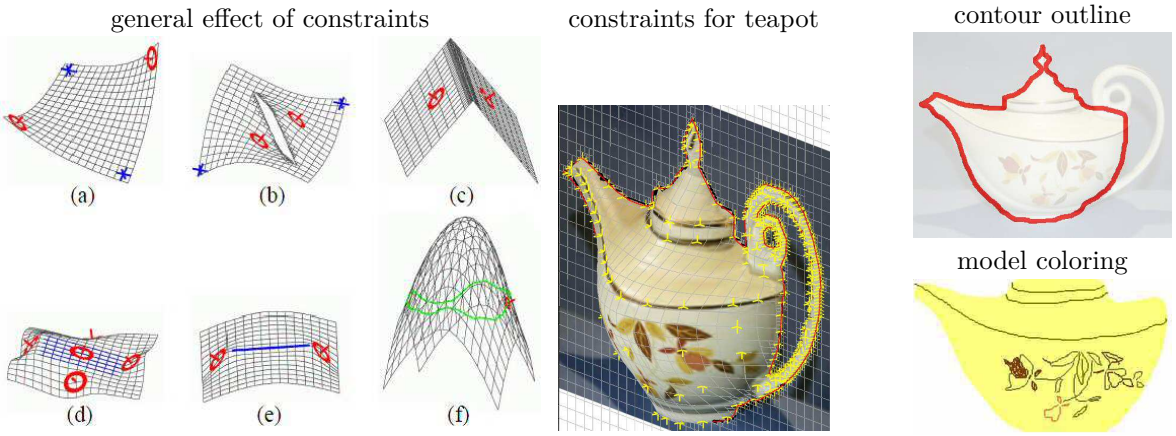
The user input of our **Shape Prior Approach** (Chapter 6) amounts to some user strokes for silhouette extraction and the adaption of the data term shape by changing parameters $k, \lambda_{\text{offset}}, \lambda_{\text{factor}}, \lambda_{\text{cutoff}}$ from Equation (6.7) which is necessary in most cases. Surface creases can be optionally added, but were not necessary for most experiments.

For both our **Volume Prior Approaches** (Chapters 7 and 8) the object shape is mainly defined by the silhouette and by adapting the object volume which can be done very quickly. However, the 2.5D approach is faster then the full 3D one, and is thus more interactive and allows for quicker modeling.

The user inputs for each method is summarized in Table 9.2. Necessary user input is printed in bold. The other inputs are either optional or the program provides a heuristic to initialize these values reasonably well.

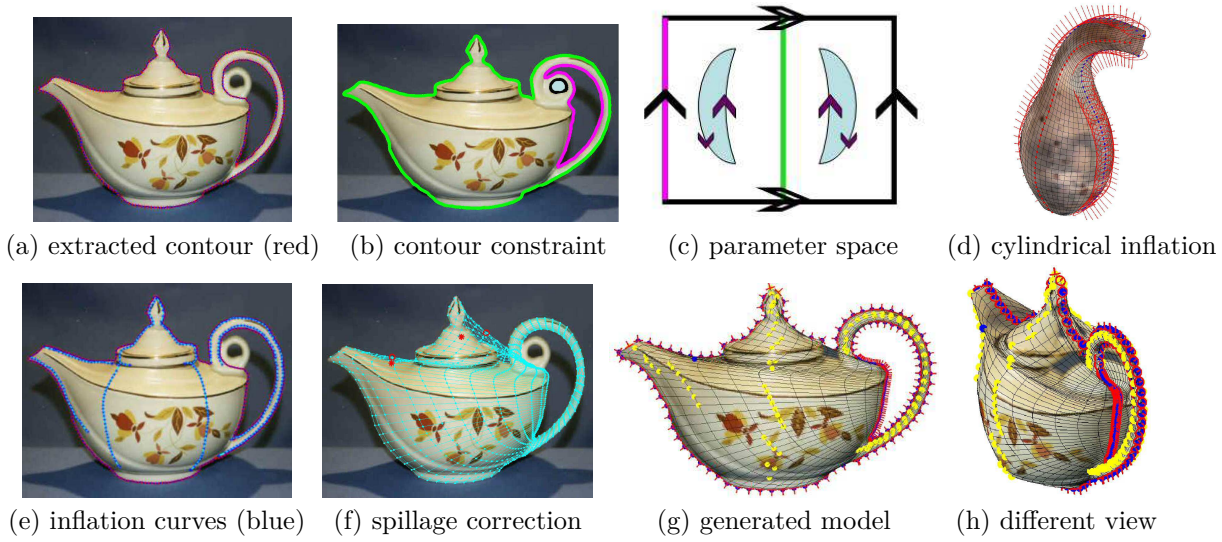
Evaluation of Experiments. The modeling process with the tool by **Zhang et al.** [253] can be cumbersome because most constraints only have a local influence on the surface shape and many of them are usually necessary. The oblique position of the donut with respect to the image plane (Figure 9.2) is difficult to model with local constraints only. Further, fine scale structures such as the teapot handle or the leg of the ostrich (Figure 9.3) are hard or impossible to model due to the limited mesh resolution. An advantage of this method is the full user control due to a variety of modeling possibilities which allows for modeling details like the round shaped eye of Dory or its side fin bending away from the fish body (Figure 9.1). Such details cannot be modeled with the other four methods in this comparison. However, the freedom in modeling incurs a larger amount of user input.

The method by **Igarashi et al.** [117] generally over-smoothes the input silhouette which can be seen in many examples, e.g the peak of the bird in the ostrich example in Figure 9.2, or



Zhang et al. [253] Igarashi et al. [117]

Figure 9.4.: User input for the methods of Zhang et al. [253] and Igarashi et al. [117]. The first image shows the general effects of different constraints and is taken from [253]. In particular, the constraints are: (a) position (blue) and normal (red) constraints, (b) depth discontinuity constraint, (c) crease constraint, (d) planar region constraint, (e) curvature minimizing fairing curve and (f) torsion minimizing fairing curve (see [253] for further details).



(a) extracted contour (red) (b) contour constraint (c) parameter space (d) cylindrical inflation

(e) inflation curves (blue) (f) spillage correction (g) generated model (h) different view

Figure 9.5.: Necessary and optional steps and user input for Prasad et al. [183]: (a) contour extraction; (b) lines of the contour have to be related to lines in the parameter space (c); (d) and (e) demonstrate different inflation heuristics; (f) during the optional spillage correction, the user can account for silhouette inconsistencies by adding further constraints; (g) and (h) show the final model and generated interpolation constraints as yellow dots. Note that (b) and (c) show a genus 2 reconstruction, while the other teapot images show a genus 1 reconstruction. All Figures are taken from [181].

the grip of teapot lid in Figure 9.3. The main advantage of this approach is the fast and intuitive modeling of geometrically simple objects. One of the drawbacks is the restricted topology, the hole in the donut example in Figure 9.2 cannot be reconstructed. A further disadvantage is the limited influence of the user during the modeling process. The fact that surface discontinuities like sharp edges are not allowed, largely decreases the class of reconstructable objects, e.g. the tail fin of Dory in Figure 9.1, the bottom parts of vase and teapot in Figure 9.3. Only very simple roundish objects like the banana (Figure 9.1), squash and orange (Figure 9.2) or the jelly bean (Figure 9.3) can be easily and reliably reconstructed.

Method	User Input (optional and **required**)
Zhang et al. [253]	• depth map dimensions • normal / position constraints • discontinuity lines (position / normal discontinuity) • planar region constraint • curvature / torsion minimizing fairing curve constraints • manual mesh-subdivision
Igarashi et al. [117]	• **rough contour lines** • union or cut operations between objects • object coloring
Prasad et al. [183]	• mesh resolution • **silhouette extraction** • **define corresponding parameter space boundaries** (defines topology) • **assign parts of the contour to lines in the parameter space** • choose **inflation heuristic** (cylindrical, cylindrical by parts, distance transform, approximation constraints) + further inflation input • spillage correction (correct silhouette consistency violated through optimization) • surface creases
Shape Prior [1] Chapter 6	• volume dimensions • **silhouette extraction** • define data term shape interactively (4 parameters) • surface creases
Volume Prior [3, 5] Chapters 7,8	• volume dimensions • **silhouette extraction** • define target volume interactively (1 parameter) • surface creases

Table 9.2.: Necessary (**bold**) and optional user inputs and modeling steps for several methods in comparison. Optional user inputs are still required algorithm inputs but they can be predefined by default values or simple heuristics and later on changed by the user if desired. Note that the variety of user input shown in this table does not reflect the amount or complexity of the input that is necessary for a reconstruction.

The main characteristic of the method by **Prasad et al.** [183] is parametric surface representation. For simple geometry such as the orange example in Figure 9.2 or the jelly bean in Figure 9.3 this facilitates the reconstruction of objects. However for objects with higher surface genus, e.g. the genus 2 teapot example (Figure 9.3), the parametrization gets sophisticated. Further, the parametrization is not uniform on the surface which makes it difficult to model elongated structures like the ostrich in Figure 9.2. This also leads to surface oscillations of the object surface as visible on the teapot handle (Figure 9.3). Another disadvantage is that silhouette consistency is not strictly enforced. Nonetheless, this method generated the most accurate results for the non-side-view examples (banana and teapot).

Single-view modeling with our **Shape Prior Approach** (Chapter 6) is mostly intuitive and many examples did not need much effort and little user experience. For instance, the banana, fish, dory (Figure 9.1), squash, orange, ostrich and donut examples (Figure 9.2) or the jelly bean (Figure 9.3) example are easy to accomplish, especially in comparison to the method by Zhang et al. [253]. In contrast to the other methods, we assume to get side-views of symmetric objects, which restricts the applicability of our method, e.g. in the donut example in Figure 9.2 the size of the hole is too small.

The results of our **Volume Prior Approaches** are fairly similar. Apart from the quicker modeling this approach shares most of the advantages and disadvantages with our shape prior approach.

Method	Advantages (+) and Disadvantages (−)
Zhang et al. [253]	+ large variety of constraints allows for flexible user modeling + user has full control of every surface detail − reconstructions are restricted to a depth map − occluded object parts cannot be modeled, synthesized views from different angles will reveal those areas − large amount of user input is often necessary − user experience and training necessary
Igarashi et al. [117]	+ very easy to use and fast interactive modeling − over-smoothes the input silhouette − smoothness properties cannot be changed by the user − not silhouette consistent − topology is limited to genus 0
Prasad et al. [183]	+ objects can also be modeled from oblique view points + apart from the silhouette the user can also use contour edges for modeling − parametric surface representation limits topology and object shape (many long elongated structures are difficult to model) − higher complexity of user input (requires expert knowledge) − silhouette consistency is not guaranteed and may require additional user input
Shape Prior [1] Chapter 6	+ moderately fast modeling + reconstructions are silhouette consistent − objects need to be symmetric, a side view is required
Volume Prior, [3, 5] Chapters 7,8	+ fast modeling + very little user input + reconstructions are silhouette consistent − objects need to be symmetric, a side view is required − user can barely influence the surface shape − limited possibilities to add surface creases

Table 9.3.: Overview of advantages and disadvantages for each method. Note that the number of advantages and disadvantages is not important in this listing since each point weights differently depending on the desired application of each method.

9.1.2.1. Summary

The theoretical and experimental comparison of the methods for curved object reconstruction identified several advantages and disadvantages which are listed in Table 9.3. In general, the performance of a method highly depends on the application. Each method has its strengths and weaknesses when applied to a specific class of single-view reconstruction problems.

The results of our experiments support the hypothesis that *generality and flexibility* of a reconstruction method is traded for the *amount of user input* or *expert knowledge*. Expert knowledge refers to the *variety* or *complexity* of the user input. The flexibility of modeling fine details with the method by Zhang et al. [253] requires the user to know and understand a variety of modeling constraints and it needs a large amount of user input. On the other hand, the method by Prasad et al. [183] needs less user input, but increases its complexity such as the definition of a suitable surface parametrization. The comparatively simple and small amount of user input for the methods by Igarashi et al. [117], and the ones proposed in this thesis (Chapters 6 to 8) comes along with the limited generality and flexibility of these methods.

9.2. Conclusion

In this chapter, we compared the single-view approaches proposed in this thesis (Chapters 6 and 7) with closely related state-of-the-art methods. Apart from a qualitative comparison on a wide variety of natural and man-made objects, we also compared important algorithm properties (see classification Section 5.2), especially the amount of user input, and highlighted advantages and disadvantages of each method.

Part III.

Spatio-Temporal Multi-View Reconstruction

10. Introduction

They always say time changes things, but you actually have to change them yourself.

Andy Warhol
(American Artist, 1928 - 1987)

In this part of the thesis we want to generalize the 3D reconstruction model by Kolev et al. [134] to the spatio-temporal multi-view case. Instead of a static scene, we now want to recover the dynamically changing surface of a moving scene based on the image sequences from a set of cameras which observe the same scene from different view points. Figure 10.1 depicts an example scene in which two children have been filmed by 16 cameras. The input images are shown next to their corresponding pre-computed camera location. Essentially, with the time domain, we simple add one more dimension to the original 3D reconstruction problem, but we will see in the following that several things change and need special consideration.

The benefit of such a generalization should of course be a reconstruction of better quality. In contrast to time-independent 3D reconstruction, the basic idea is to make use of additional input information from other time steps to improve the accuracy of the reconstruction by enforcing some kind of temporal coherence. If 3D reconstruction is performed separately for each time step, the location of the reconstructed surface might change drastically and incorrectly over time in areas with low, missing or ambiguous input information. Moreover, for

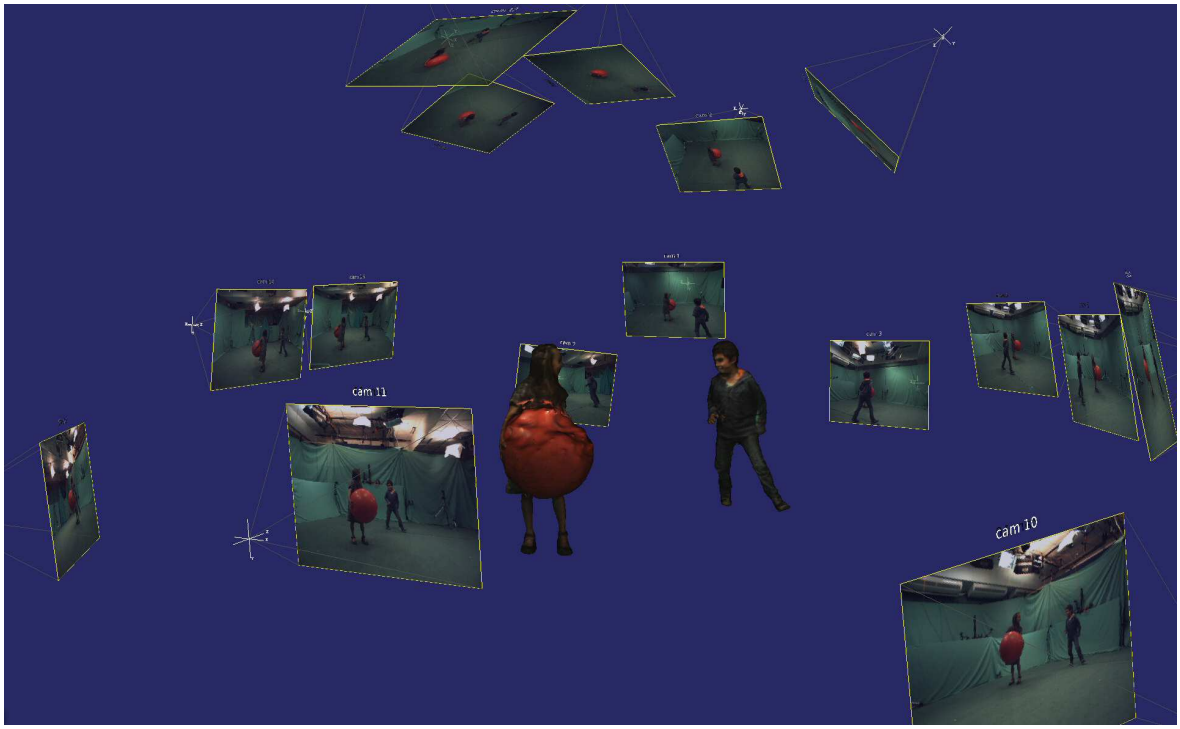


Figure 10.1.: Spatio-temporal multi-view reconstruction of a dynamic scene from 16 simultaneous input sequences visualized at their corresponding camera positions. ('Children playing' scene from [121]).

non-artificial input data, the camera images will always contain noise which also leads to noise in the reconstructed surface. This noise becomes visible as a jittering of the reconstructed surface when played over time.

Compared to the typical scenario of multi-view 3D reconstruction of static scenes, a generalization to a spatio-temporal setup brings up several **practical challenges**:

- **Fewer cameras.** Video cameras are usually more expansive than photo cameras which results in setups with viewer cameras and thus wider average baseline. Many existing 3D reconstruction approaches rely on small baselines for photometric matching and the reconstruction quality drops significantly with fewer cameras or breaks down completely.
- **Lower resolution.** Video cameras typically have a lower resolution than photo cameras, which results in coarser reconstructions.
- **More camera noise.** Video cameras usually possess a higher noise level than photo cameras which leads to lower photometric matching scores.
- **Temporal camera noise.** The noise pattern of the camera images changes over time and leads to different depth estimates in every time step even if both the camera position and the scene is static.
- **Motion blur.** Depending on the motion speed of the dynamic scene and exposure settings of the camera, image parts observing fast motion might be substantially blurred which makes the photometric matching more difficult.
- **Camera sychronization.** In contrast to static scenes, the cameras need to be temporally synchronized in order to perform any photometric matching.
- **High demands on memory and computation time.** Since cameras usually acquire images at 25-30 frames per second, the amount of input data, even for short video sequences, is very large. Further, the demands on memory and computation time of the reconstruction algorithm increase substantially if one wishes to exploit the temporal coherence of the reconstruction over consecutive time frames.

We will approach these problems in the next Chapter 11 by extending our 3D reconstruction framework to favor temporal consistency.

10.1. Problem Statement and Notation

We now formulate the problem of spatio-temporal mutli-view reconstruction as finding a minimizer of a variational minimal surface energy. Following the idea of the first variational approach to multiple view 3D reconstruction by Faugeras and Keriven [79], we want to find a surface $\Sigma \in \mathbb{R}^3$ which minimizes the photometric error.

Similar to the previous Part II we will make use of the weighted total variation (Definition 2.13) and extend the 3D reconstruction model in Equations (3.11) and (3.12) in the following way.

$$\Sigma^* \in \arg\min_{\Sigma} \left\{ \int_{\Sigma} \rho \, ds + \int_{int(\Sigma)} f \, d\boldsymbol{x} \right\} , \tag{10.1}$$

where $int(\Sigma)$ denotes the interior of the surface (and its boundary is again the surface: $\partial int(\Sigma) = \Sigma$). The function ρ locally weights the surface area in a geodesic manner and will encode the photometric matching data. This way the surface will "snap" to locations with high photoconsistency because the corresponding surface area is locally decreased.

As described in Chapter 3, the definitions and properties for minimal surfaces hold for any

dimension and we can simply define surface Σ to be a 3-dimensional hypersurface embedded in the 4-dimensional spatio-temporal space. By transforming the reconstruction problem in Equation (10.1) via an indicator function u on a higher dimension, we lift it from 3D to the spatio-temporal (4-dimensional) domain $u \in V \times T \subset \mathbb{R}^3 \times \mathbb{R}_{\geq 0}$ with $u = \mathbf{1}_{int(\Sigma)}$

$$u^* \in \underset{\mathcal{BV}(V \times T, \{0,1\})}{\arg\min} \left\{ \int_{V \times T} \rho \, |\nabla u| \; d\boldsymbol{x} + \int_{V \times T} f \cdot u \; d\boldsymbol{x} \right\} \; , \tag{10.2}$$

in which $\rho : \mathbb{R}^3 \to \mathbb{R}$ is a normalized measure of photometric consistency (abbreviated as photoconsistency) between the input images. Now, we still have to make sense of the 4-dimensional gradient norm, because and isotropic penalization of spatial and temporal dimensions might not be meaningful. We will define the regularization term more precisely later on the respective chapters, but Equation (10.2) essentially reflects the main idea of our approach.

To make the reconstruction problem tractable, we will impose the following **assumptions**:

- **Synchronized image sequences.** We do not deal with camera synchronization in this work.
- **Calibrated cameras.** We do not deal with camera calibration in this work.
- **Lambertian-like light model.** We use one of the simplest light models.
- **Silhouettes.** They significantly help to recover geometry in sparse camera setups.
- **Smooth surface.** The minimal surface prior favors smooth surfaces.

Silhouettes. As silhouettes we define binary images $S : \Omega \subset \mathbb{R}^2 \to \{0, 1\}$ which separate one or several objects of interest from the background in an input image. They are usually obtained with some kind of segmentation algorithm (e.g. [189, 219]) or, more typical for videos, with background subtraction techniques (see e.g. [171, 207, 21]). We assume to have one silhouette for each corresponding input image. We will mainly use the silhouettes in order to reduce the solution space by projecting and intersecting all available silhouettes in the 3D space which leads to the visual hull concept.

Visual Hull. Visual hulls unify the information gained from an object silhouette to the multi-view case. In his PhD thesis, Baumgart [19] introduced an algorithm which calculates the intersections of projected silhouettes in 3D space. Later on Laurentini [148] coined the term "visual hull" and studied the problem in several publications.

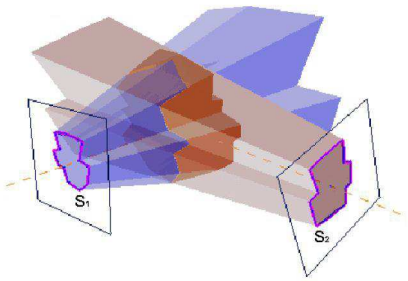


Figure 10.2.: Visual Hull as the intersection of silhouette pre-images (picture from [25]).

Definition 10.1 (Visual Hull)**.** *Let $\{\pi_i\}_{i=1}^N$ be the projection matrices of N cameras observing a scene and let $\{S_i\}_{i=1}^N$ be a set of corresponding silhouette images with $S_i : \Omega \to \{0, 1\}$. Then, the visual hull $\mathcal{VH} : V \subset \mathbb{R}^3 \to \{0, 1\}$ is defined as the intersection of the silhouette pre-images from all cameras:*

$$\mathcal{VH} = \bigcap_{i=1}^{N} \pi^{-1}(S_i) \; , \tag{10.3}$$

where $\pi^{-1}(S_i)$ denotes the pre-image (or "unprojection") of silhouette S_i into the 3D domain.

The projection matrices π_i represent the projective pinhole camera model for the rest of this thesis (see [107, page 153ff] for details).

For correctly segmented silhouettes, the visual hull is the largest volume that is consistent with the silhouettes and contains the true scene. In this sense, it is a conservative estimate of the scene geometry as it misses any holes and concavities that are not visible in the silhouettes. The definition of the visual hull can be easily extended to the spatio-temporal case by adding temporal indices to the silhouettes and projection functions.

Later on, in Chapter 13, we will also use the visual hull to analyze its topology and to define topological constraints for the reconstruction process.

10.2. Related Work

In this section, we give an overview of related and important works on spatio-temporal multi-view reconstruction. Mostly, the basis for such a method is a working 3D reconstruction method. Therefore, before we discuss spatio-temporal reconstruction approaches, we also discuss some selected works on static 3D reconstruction.

10.2.1. Related Work on Multi-view Stereo Reconstruction

Static multi-view stereo reconstruction has been a central research fields since decades and a huge of amount of research paper exist. This subsection only gives a brief overview by naming a selection of important works on that topic. For more information a good overview is provided by Seitz et al. [195] and the accompanying benchmark website[1] which usually contains the most recent advances in the field.

Silhouettes-based approaches. The simplest algorithms for 3D reconstruction from silhouettes are algorithms based on the concept of the visual hull [19, 148, 25, 27, 84, 85, 86, 101].

Space carving. Instead of silhouettes Seitz and Kutulakos [141] used a voxel-based photometric matching score and defined the *photo hull* which is computed by iteratively carving voxels with high photometric error - after starting with a fully occupied scene. This idea has then been extended in several ways, e.g. [196, 35]. Generally space carving is very sensitive to outliers, because incorrectly carved surface voxels can lead to deep holes in the model due to the low photoconsistency scores of non-surface voxels.

Point cloud and mesh-based approaches. Furukawa et al. [89, 91] compute oriented point clouds based on image feature matching. In an iterative process new 3D points are expanded next to existing ones and possibly filtered due to inconsistencies with respect to the photometric matching score and the visibility. The output is a "dense" oriented point cloud which can be meshed with other methods, e.g. the popular Poisson surface reconstruction method by Kazhdan et al. [129]. Although the approach relies on local optimization and uses many heuristics it performs well on several benchmarks and is thus still considered to be among the state-of-the-art methods. Goesele et al. [93] went for larger scales and built a system for reconstructing 3D scenes from internet photo collections. Similarly, Vu et al. [234] proposed a multi-view stereo approach for large-scale reconstructions. Jancosek and Pajdla [122] propose a weighting scheme to better reconstruct weakly supported surfaces. The method extends the one by Labatut et al. [142] which computes Delaunay tetrahedra on point clouds and

[1] Middlebury Multi-view Stereo evaluation benchmark: http://vision.middlebury.edu/mview/eval/

determines their occupancy label within a graph-cut framework. Generally, the extension of point cloud-based approaches to the temporal domain is difficult, because some kind of temporal correspondence needs to be estimated to constrain the temporal coherence.

Volumetric approaches. An interesting volumetric and entirely probabilistic approach for scene reconstruction is the one by Calakli et al. [39] which is based on the model by Pollard et al. [176] developed for 3D change detection. This model been made efficient by using octrees in [64] and forms has now recently been extended to the 4D domain by Ulusoy et al. [215].

Kolev et al. [135, 136, 134] proposed the convex variational minimal surface approach for multi-view 3D reconstruction that has been derived in Chapter 3 and forms the basis of this thesis. The main advantage of this approach in 3D reconstruction is the natural way of surface regularization in 3D space, compared to stereo-based approach which usually regularize depth or disparity discontinuities. Since the method also handles arbitrary topologies, is easily extendible and globally optimizable it unifies many desirable properties. Variants of this approach have also been used in by Ummenhofer and Brox [216] for combined 3D reconstruction and camera pose estimation and by Häne et al. [106] for joint 3D reconstruction and class segmentation.

As said in the beginning of this subsection the number of static 3D reconstruction approaches is enormous. Generally, the generalization of 3D reconstruction techniques to the spatio-temporal reconstruction from videos is by no means straightforward.

10.2.2. Related Work on Spatio-temporal Multi-view Stereo Reconstruction

Silhouettes-based approaches. The visual concept is easily extendible to the temporal domain and because of its simplicity there exist many works on that topic e.g. [22, 146, 100] For 4D reconstruction these approaches are still the basis for the current state of the art in commercial products[2]. An interesting scene representation based on 4D Delaunay meshes has been proposed by Aganj et el. [12] in which every time frame corresponds to 3D Delaunay meshes with occupancy flags for each cell. The advantage of this surface representation is that Delaunay also defines a correspondence of vertices between time steps which makes linear interpolation between times frames straightforward. Generally, pure silhouette based approaches cannot recover object concavities which are not visible in the silhouettes. For a good surface approximation usually a higher number of input views is necessary as the surface approximation is coarse for a small number of cameras.

Later, in [11] Aganj et el. extended their silhouette-based approach. Rather than using silhouettes they start with a 4D point cloud computed by feature matching in every time step. After tessellating the point cloud into 4D Delaunay pentatopes each of them is then label as occupied or empty by means of a globally optimal graph-cut approach. Although the scene representation has several attractive properties the main drawback of this method is the point cloud generation step, which is prone to noise and outliers.

Wuermlin et al. [243] used dynamic point samples in space-time for real-time free-viewpoint video. This approach is merely a view point interpolation system which uses splatting techniques for novel view point rendering.

2.5D+time approaches. As one of the first works, Zhang et al. [252] extended the problem of classical binocular stereo matching to the spatio-temporal domain. Guillemaut and Hilton [102] jointly solve the problem of multi-layer segmentation and depth estimation within

[2] see for example http://www.4dviews.com

a graph-cut framework. They enforce temporal coherence by means of optical flow measures which are weighted according to their confidence to account for unreliable flow estimates. Richardt et al. [186] proposed a method for spatio-temporal filtering and upsampling of RGB-Depth videos. Generally, these approaches mostly rely on the properties of their 2.5D surface representation and their generalization to full 3D is usually not straightforward. Also in order to obtain full 3D models an additional merging step of several depth maps is necessary.

Volumetric approaches. Pioneering work on the topic of spatio-temporal multi-view 3D reconstruction in a continuous setting has been done by Goldluecke et al. [94, 95, 97]. They described the evolution of a space time surface by means of level set functions which iteratively approach a local minimum of the respective energy. Generally, level set methods rely on a proper initialization to converge to the desired solution due to the locally optimal optimization procedure. Starck and Hilton [201] proposed a spatio-temporal reconstruction pipeline which first estimates shapes from silhouettes and later refines the reconstruction with photometrically matched features and information about the reconstruction result from the previous time step with volumetric graph-cuts.

Generally, volumetric approaches have proved to be suitable for spatio-temporal reconstruction, because temporal alignment can be expressed similarly as spatial alignment. The spatio-temporal reconstruction approaches proposed in this thesis also belong to this category.

Scene flow approaches. The term *scene flow* often refers to 2.5D approaches that the estimate motion of the maps over time, e.g. Wedel et al. [236], or Vogel et al. [233]. Since we are interested in full 3D models the following approaches which compute a 3D scene flow are more related to this work. Vedula et al. [224] compute the 3D dimensional scene flow based on the 2D optical flow from several input videos. Guan et al. [99] compute a dense volumetric occupancy flow from silhouettes within a probabilistic framework which is solved by a locally optimal expectation-maximization optimization.

Combined scene flow and 3D reconstruction approaches. Vedula et al.[225] generalized space carving approach to the spatio-temporal domain by defining a photoconsistency measure in 6D space. The jointly estimate motion and geometry by rejecting (carving) inconsistent ones. The work by Neumann and Aloimonos [164] is one of the early works that estimated the 3D geometry and its movement jointly over time. Using a multi-resolution subdivision surface representation they jointly estimate the position and motion of each vertex over time. Vedula et al.[223] estimate 3D scene flow from multiple optical flows for spatio-temporal view point interpolation. The quality of this approach is limited the view point interpolation is done via ray-casting on an approximate surface proxy that has been fitted to space-carved voxel model. In [179], Pons et al. proposed a variational method for combined stereo reconstruction non-rigid motion estimation and presented later in [177] a more general method for combined multi-view stereo reconstruction non-rigid motion estimation from multiple video sequences. In a later work, Pons et al. [178] jointly estimate scene geometry and scene flow by minimizing the reprojection error in a locally optimal coarse-to-fine approach. Sharf et al. [198] study the problem of space-time reconstruction by means of incompressible flow. Courchay et al. [60] propose a mesh-based approach with fixed vertex connectivity imposing limitations in cases where objects join or separate or change their topology. Such cases need special consideration and are non-trivial in the general case. Tung et al. [213] do not compute scene flow, but incorporate optical flow of the input sequences as additional features which are fused in a probabilistic Markov random field framework for spatio-temporal 3D reconstruction.

Model-based approaches. The idea behind model-based approaches is to treat the two problems of reconstructing geometry and estimating their motion separately, because it usually simplifies the problem. The geometry model only needs to be acquired once in advance with any 3D reconstruction algorithm or by using handcrafted models. Computing its motion can then be seen as a tracking or model alignment problem which usually has much less unknowns then the original joint estimation problem. The simplicity of this approach is bought by giving up flexibility: 1) Every object in the scene needs to be acquired in advance usually under controlled conditions to ensure all necessary details are captured. 2) It is more difficult to handle topology changes of the models over time. 3) Due to the reduced number of parameters the models usually cannot recover smaller details of the deformation. A classical example for this issue are foldings and wrinkles in clothing that cannot be tracked correctly when recovering the geometry of humans over time. Although these approaches use the same input data they address and solve a slightly different problem, which is mostly body pose estimation. Nevertheless, we want to mention a few important works in this subfield.

Furukawa et al. [90] capture a model via reconstruction based on their PMVS-3D reconstruction approach [89], then the polyhedral mesh with fixed topology is propagated by tracking its vertices. Pose estimation and articulated mesh animation from silhouettes have been investigated by Vlasic et al. [229]. They further deform the shape of the mesh to fit the input silhouettes after the pose estimation for each frame. De Aguiar et al. [69] do not use skeleton models or pose estimation, but directly estimate 3D vertex correspondences and use a Laplacian mesh deformation scheme to align high-quality meshes. This approach is better suited for larger model deformations, e.g. when tracking people wearing wide clothing. They further improved the approach by combining surface-based with volumetric deformation techniques [68]. Varanasi et al. [222] identify sparse, but robust matching vertices in consecutive time frames based on geometric and photometric information and then densify the motion field with Laplacian diffusion. Cagniart et al. contributed a patch-based approach [37] and a probabilistic approach [38] to deformable mesh tracking.

11. Spatio-Temporal Multi-View 3D Reconstruction

Essentially, all models are wrong, but some are useful.

George Edward Pelham Box
(Professor em. of Statistics, University of Wisconsin, 1919 - 2013)

11.1. Introduction

In this chapter, we generalize the variational convex 3D reconstruction approach by Kolev et al. [135] to the spatio-temporal domain (see Figure 11.1). The global optimality of the approach and especially the more natural regularization in the 3D domain rather than in the image domain makes the approach attractive. Although a variety of useful regularizers for depth maps have been presented in the literature, intuitively they do not provide a good regularization in a multi-view setup because we are usually looking for a connected and locally smooth surface rather than a smooth depth map. 3D reconstruction based on depth maps is a popular approach to this problem and many works exist on this topic e.g. [248],[122]. Inherently these approaches split the overall problem into two separate ones: depth reconstruction followed by surface reconstruction based on these depth maps. As a result, important information such as the consistency of an estimated depth map value is usually not handed over into the following surface reconstruction. In contrast, our goal is to carry as much information as possible into the final global 3D surface optimization.

Apart from the work by Kolev et al. [135], our approach is also related to the space-time 2D tracking framework by Unger et al. [218]. They cast the problem of tracking objects in images over time as a 3D segmentation problem to model temporal smoothness or deal with temporally short occlusions of the tracked object. Although the task and several properties are quite different we use a similar model, but in a 4D rather than a 3D setting.

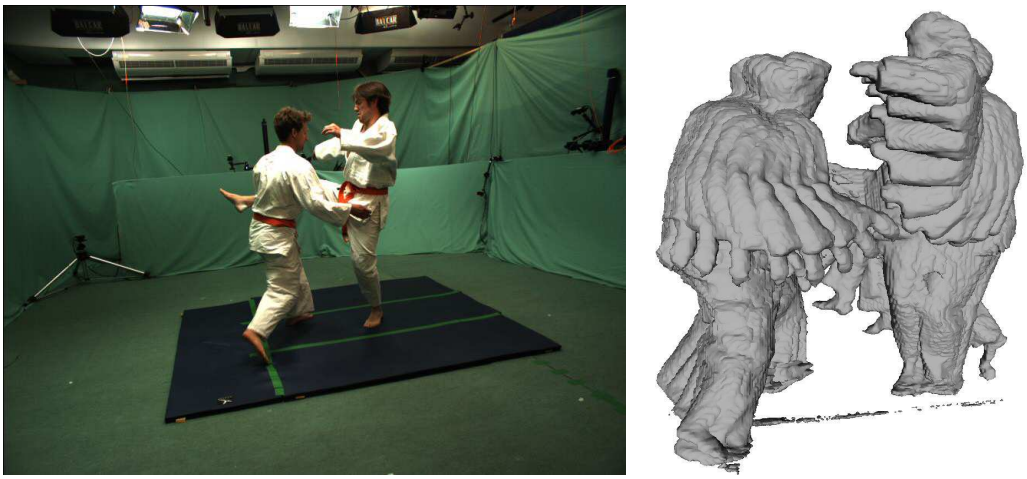


Figure 11.1.: One of the input images and several time frames of a space-time surface evolution.

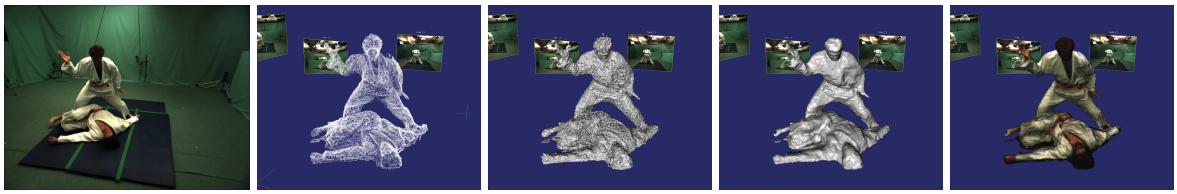


Figure 11.2.: Outline of the proposed space-time reconstruction framework. Two men are filmed synchronously by 16 cameras. The figure shows (left to right) one input image, estimated photoconsistencies, a level set of the proposed data term, the final reconstructed mesh shaded and textured.

Contributions.

- We generalize the works of Unger et al. [218] and Kolev et al. [135] from the three-dimensional setup to a four dimensional one leading to a mathematically transparent and globally optimal approach for space-time multi-view 3D reconstruction.
- In order to make the 3D reconstruction approach by Kolev et al. [135] work in wide-baseline camera setups we propose a novel data term, which has several desirable properties and improves the one in [135] in several aspects. Firstly, it better preserves surface edges and concavities. Secondly, it has better hole filling abilities when photoconsistency information is weak and sparse. Finally, it does not have a global influence, that is, it does not affect surface parts which are not visible in the respective camera.
- Further, we reduce the computation time per frame from several hours, as reported by [135], to about 1-2 minutes for equivalent volume sizes. This aspect is important when processing longer sequences.

In the following we introduce our space-time reconstruction framework which is outlined in Figure 11.2. We then explain how to compute respective terms. In Section 11.3 we discuss the optimization procedure and give some details on the implementation in Section 11.4. Section 11.5 presents results on several data sets and Section 11.6 concludes the chapter.

11.2. Variational Space-Time Reconstruction

Let $V \subset \mathbb{R}^3$ describe a volume in space and let $T \subset \mathbb{R}_{\geq 0}$ represent the temporal domain. We are looking for a smooth hypersurface Σ in the space $V \times T$ which best explains the series of input images with known projections $\{\pi_i\}_{i=1}^N$. For ease of notation we will drop the temporal index whenever the meaning is clear by the context. Similar as in [135] we represent surface Σ by means of a binary labeling function $u : V \times T \to \{0, 1\}$ which indicates surface interior (1) or exterior (0). We follow the path of their work and define an energy function which measures both the surface smoothness and how well the surface fits to the input data.

$$E(u) = \int_{V \times T} \left(\rho |\nabla_{\boldsymbol{x}} u| + g_t |\nabla_t u| \right) d\boldsymbol{x} dt + \lambda \int_{V \times T} f u \; d\boldsymbol{x} dt \tag{11.1}$$

The data term f in the second term of Equation (11.1) gives local preferences for either an interior or an exterior label and will be defined in Section 11.2.2. It is weighted by parameter $\lambda > 0$ to favor either a smooth surface or a surface that aligns with the potentially noisy data. The task of the first term - the regularization term - is to reject outliers, deal with locations of missing data and to favor a spatially and temporally smooth surface. To account for the inherent difference between spatial and temporal dimensions this term is split into a spatial and a temporal part which then regularizes these dimensions in an anisotropic manner.

The spatial regularization is weighted by function $\rho : V \times T \to \mathbb{R}$ which represents the photo-consistency measure being defined in the following section. Weighting down the penalization of the gradient norm ρ makes the surface boundary snap to probable surface locations which are indicated by a low value of ρ. Note that for this reason, there is an inverse relation between a high photoconsistency corresponding to a low value of ρ and vice versa.

In Equation (11.1) function $g_t : V \times T \to \mathbb{R}$ steers the temporal smoothness. We choose it as a function that depends on the gradient magnitude of the data term:

$$g_t(\boldsymbol{x}, t) = \exp\big(-a|\nabla_t f(\boldsymbol{x}, t)|^b\big) \ . \tag{11.2}$$

This choice of $g_t(\cdot)$ prevents locations with strong gradients from being over-smoothed which is a favorable property in the presence of fast surface motions. The purpose of the temporal regularization is mainly to suppress temporal noise in the surface reconstruction rather than penalizing surface motion in a dynamic scene. The effects of parameters a and b will be discussed in the experimental section.

11.2.1. Photoconsistency Estimation

For every time step and for each camera i we define a cost function[1] $C_i : V \times \mathbb{R} \to \mathbb{R}$ which calculates a matching cost at a location defined by distance d from the camera center towards or through point $\boldsymbol{x}$ based on the normalized cross correlation (NCC)

$$C_i(\boldsymbol{x}, d) = \sum_{j \in \mathcal{C}' \setminus i} w_i^j(\boldsymbol{x}) \cdot \mathrm{NCC}\Big(\pi_i\big(r_i(\boldsymbol{x}, d)\big), \pi_j\big(r_i(\boldsymbol{x}, d)\big)\Big) \ . \tag{11.3}$$

The function $r_i : V \times \mathbb{R} \to V$ returns points on the ray from camera i through point $\boldsymbol{x}$ according to a given distance d from the camera and π_i, π_j are the projection matrices for cameras i and j. The normalized cross-correlation function $\mathrm{NCC} : \Omega_i \times \Omega_j \to [0, 1]$ measures the discrepancy between two normalized image patches in cameras i and j. Let $\bar{I}(p) = \sum_{q \in \mathcal{N}(p)}$ be the mean patch color and $\widetilde{I}_i = I(q_i) - \bar{I}(p_i)$ be the color difference of neighbor pixel q_i and the mean patch color at pixel p_i for invariance against additive lighting changes then we use the zero mean normalized cross correlation as

$$\mathrm{NCC}(p_i, p_j) = \sum_{(q_i, q_j) \in \mathcal{N}(p_i, p_j)} \frac{\widetilde{I}_i \cdot \widetilde{I}_j}{\sqrt{\frac{1}{|\mathcal{N}(p_i)|} \sum_{q_i \in \mathcal{N}(p_i)} \widetilde{I}_i^2} \cdot \sqrt{\frac{1}{|\mathcal{N}(p_j)|} \sum_{q_j \in \mathcal{N}(p_j)} \widetilde{I}_j^2}} \ , \tag{11.4}$$

where $\mathcal{N}(p_i)$ defines the image patch around p_i as a local neighborhood of $p_i \in \Omega$. To calculate $C_i(\cdot)$ we select a subset of front-facing cameras $\mathcal{C}' \subset \mathcal{C}$ for which the angle between the viewing directions is below γ_{max}=85°. The contribution of each camera is weighted by a normalized Gaussian weight $w_i^j(\boldsymbol{x})$ of the angle between view directions of cameras i and j. Further, we discard unreliable correlation values by means of a threshold $\tau_{\mathrm{NCC}} = 0.3$ and truncate C_i to zero by setting

$$\bar{C}_i(\boldsymbol{x}, d) = \begin{cases} 0, & \text{if } C_i(\boldsymbol{x}, d) < \tau_{\mathrm{NCC}} \\ C_i(\boldsymbol{x}, d), & \text{otherwise} \end{cases} \tag{11.5}$$

This prevents $C_i(\cdot)$ from being negative and the truncation to zero will lead to a neutral behavior for its use in the regularizer as well as in the data term. For the photoconsistency

[1] The temporal dependency is omitted for better readability.

measure ρ we employ the voting scheme of Hernández and Schmitt [77]

$$\rho(\boldsymbol{x}, t) = \exp\Big[-\mu \sum_{i \in \mathcal{C}'} \underbrace{\delta\big(d_i^{\max} = \mathrm{depth}_i(\boldsymbol{x})\big) \cdot \bar{C}_i(\boldsymbol{x}, d_i^{\max})}_{\mathrm{VOTE}_i(\boldsymbol{x})} \Big] \tag{11.6}$$

which accumulates votes from different cameras only in locations $\boldsymbol{x} \in V$ if the maximum quality along the ray through the center of camera i and $\boldsymbol{x}$ is found at distance $d_i^{\max} = \arg\max_d \bar{C}_i(\boldsymbol{x}, d)$. Thus, every camera ray has exactly one measurement if the corresponding matching score exceeds the threshold. Function $\mathrm{depth}_i : V \to \mathbb{R}$ returns the Euclidean distance of $\boldsymbol{x}$ to the center of camera i. The scaling parameter has been set to $\mu = 0.15$. Function $\rho(\cdot)$ represents a matching score of how well a small surface patch in $\boldsymbol{x}$ matches both corresponding camera images. It thus indicates probable surface locations with a low value. In the next section we explain how this information can be used for a proper modeling of the data term.

11.2.2. Data Term for Multi-View Reconstruction

The data term is necessary to avoid trivial solutions when minimizing Equation (11.1) and replicates photoconsistency information in form of local labeling preferences. In a multi-view setup, each label of $u(\boldsymbol{x})$ depends on the labels of all points along all the camera rays passing through $\boldsymbol{x}$. Considering these dependencies accurately generally leads to an involved non-convex optimization problem. We argue that these dependencies can be approximated by means of unary potentials f. Negative values of f favor an interior label, while positive ones an exterior label of u. The photoconsistency measure defined in the last section gives hints about probable surface locations. However, it is not directly usable to express regional affinity. Our goal is to carry the uncertainties about the surface location indicated by quality functions $C_i(\cdot)$ into the unaries f and thus into the global optimization of energy (11.1). We assume that the maximum-filtered NCC score at point $\boldsymbol{x}$ has the following relation to the probability that surface Σ passes through this point:

$$P_i(\boldsymbol{x} \in S) = 1 - \frac{1}{Z} \exp\Big[-\eta \cdot \mathrm{VOTE}_i(\boldsymbol{x}) \Big] \tag{11.7}$$

where Z is a normalization constant. Parameter η steers the exponential relationship between the number of cameras giving a vote, their corresponding voting qualities $\mathrm{VOTE}_i(\boldsymbol{x})$ and the probability that the point $\boldsymbol{x}$ is part of the surface. Each camera ray may give a single vote for a probable surface location. Starting from this location and walking towards the respective camera i we follow the idea that each time we pass another probable surface location, the probability of being in the surface interior further decreases. This idea is expressed in the following equation which defines the probability of point $\boldsymbol{x}$ being in the surface interior for a reference camera i:

$$P_i(\boldsymbol{x} \in int(S)) = \prod_{j=1}^{N} \prod_{\mathrm{depth}_i(\boldsymbol{x}) < d \leq d_i^{\max}} \Big[1 - P_j\big(r_i(\boldsymbol{x}, d) \in S\big) \Big] \tag{11.8}$$

The inner product integrates the surface probability votes along the ray between $\mathrm{depth}_i(\boldsymbol{x})$ and $d_i^{\max}$ and the outer product accounts for the fact that these probabilities come from other cameras. We assume independence of individual cameras and obtain the overall probability that $\boldsymbol{x}$ is an interior point:

$$P(\boldsymbol{x} \in int(\Sigma)) = \prod_{i=1}^{N} P_i(\boldsymbol{x} \in int(\Sigma)) \tag{11.9}$$

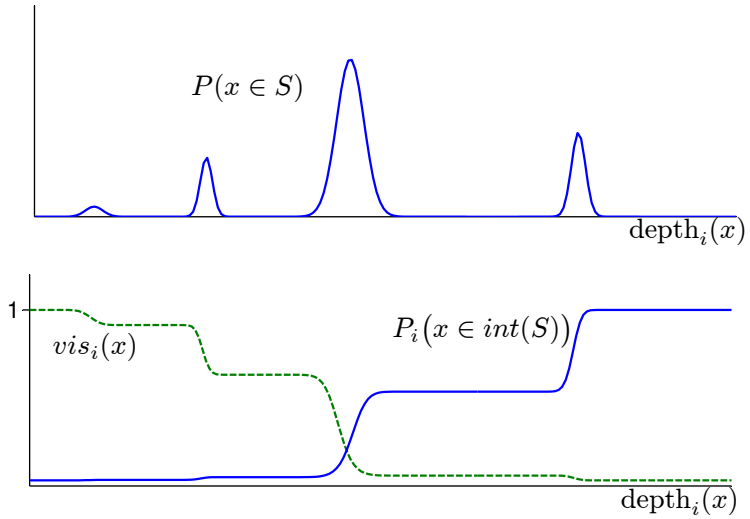


Figure 11.3.: Schematic plots of probabilities along a camera ray. The center of camera i is in the coordinate origin. $P_i\big(\boldsymbol{x} \in int(S)\big)$ and $vis_i(\boldsymbol{x})$ multiplicatively integrate the probabilities $P(\boldsymbol{x} \in S)$ along the ray before and behind location $\boldsymbol{x}$ respectively (when looking from the camera).

Finally we define data term f in Equation (11.1) as the log-probability ratio:

$$f(\boldsymbol{x}, t) = -\ln\left(\frac{1 - P\big(\boldsymbol{x} \in int(S)\big)}{P\big(\boldsymbol{x} \in int(S)\big)}\right) \quad . \tag{11.10}$$

Equation (11.8) is related to the probabilistic visibility model used by Pollard and Mundy [176, Eq.(4)]. They define the visibility $\text{vis}_i(\boldsymbol{x})$ of a point $\boldsymbol{x}$ as the probability that $\boldsymbol{x}$ is not occluded by any other point between $\boldsymbol{x}$ and the camera center:

$$\text{vis}_i(\boldsymbol{x}) = \prod_{0<d<\text{depth}_i(\boldsymbol{x})} \Big[1 - P_i\big(r_i(\boldsymbol{x}, d) \in S\big)\Big] \tag{11.11}$$

One could argue that $1 - \text{vis}_i(\boldsymbol{x})$ is also a good indicator for being in the surface interior. However, as long as none of the $P_i(\boldsymbol{x} \in S)$ equals exactly one, $\text{vis}_i(\boldsymbol{x})$ never reaches zero and will influence the probability of $\boldsymbol{x}$ being inside the surface far behind the camera vote. This model propagates the uncertainty that a ray from the camera center has passed a surface forward infinitely into the scene. In contrast, we propose a more conservative approach: we propagate the uncertainty of a ray-surface intersection from the local camera vote only towards the respective camera centers. This way the uncertainty is only distributed in between the camera and the location of its vote. Figure 11.3 illustrates the shape of these probability distributions schematically. Visually speaking, every camera vote carves its way towards the camera with its corresponding probability measure and the multiplication of all such camera bundles gives the probability of being in the surface interior. As a desirable result, this approach does not influence areas where photoconsistency information is missing. This way the data term favors the photo hull wherever photoconsistency information is missing or unreliable. Note that we do not need to assume any minimal surface thickness as it is usually done in approaches dealing with truncated signed distance functions (e.g. [248]). In contrast to the data term proposed in [135] our approach does not influence the estimates of other surfaces behind the camera vote.

11.3. Global Optimization

To minimize energy (11.1) we relax the image of function u to $[0,1]$ and employ the preconditioned primal-dual algorithm by Pock and Chambolle [175] - see Section 4.3.4. Equation (11.1) can be rewritten by introducing a dual variable $\boldsymbol{p} : V \times T \rightarrow \mathbb{R}^4$ that helps to deal with the non-differentiability of the total variation norm. The derivations follow the ones of Unger et al. [218]:

$$E(u) = \max_{\|\boldsymbol{p}\| \leq 1} \int_{V \times T} \langle u, -\operatorname{div}(\boldsymbol{p}) \rangle \; d\boldsymbol{x}dt + \lambda \int_{V \times T} fu \; d\boldsymbol{x}dt \tag{11.12}$$

This saddle point problem is optimized by means of an iterative update scheme performing a gradient ascent in the dual and a gradient descent in the primal variable:

$$\begin{aligned} \boldsymbol{p}^{k+1} &= \Pi_C \left[\boldsymbol{p}^k + \sigma \nabla \bar{u}^k \right] \\ u^{k+1} &= \Pi_{[0,1]} \left[u^k + \tau (\operatorname{div}(\boldsymbol{p}^{k+1}) - \lambda f) \right] \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k \end{aligned} \tag{11.13}$$

The projection Π of u onto the unit interval $[0,1]$ is done by thresholding. Projection onto the set $C = \{q = (q_{\boldsymbol{x}}, q_t)^T : V \times T \rightarrow \mathbb{R}^4 | \; \|q_{\boldsymbol{x}}\| \leq 1, |q_t| \leq 1\}$ is a projection on a 4D hyperball and can be done as follows:

$$\Pi_C(q) = \left(\frac{q_{\boldsymbol{x}}}{\max(1, \frac{\|q_{\boldsymbol{x}}\|}{\rho})}, \max\left(-g_t, \min(g_t, q_t) \right) \right)^T \tag{11.14}$$

The step sizes σ and τ are chosen adaptively by keeping track of the corresponding operator norms as suggested in [175] for diagonal preconditioning. For the primal variable u we assume von Neumann boundary conditions for both spatial and temporal derivatives and corresponding Dirichlet boundary conditions for $\boldsymbol{p}$, that is $\nabla u \big|_{\partial(V \times T)} = 0$ and $\boldsymbol{p} \big|_{\partial(V \times T)} = 0$. The update scheme in Equation (11.13) provably converges to a global minimum of relaxed energy (11.1). The corresponding optimal binary labeling can be found by simple thresholding of the relaxed solution [175].

11.4. Implementation

Both the photoconsistency estimation as well as the energy optimization have been implemented on the GPU using the NVidia CUDA framework. The optimization scheme in Equation (11.13) lends itself to a parallel implementation. In the result section we also briefly detail the implementation of the photoconsistency estimation.

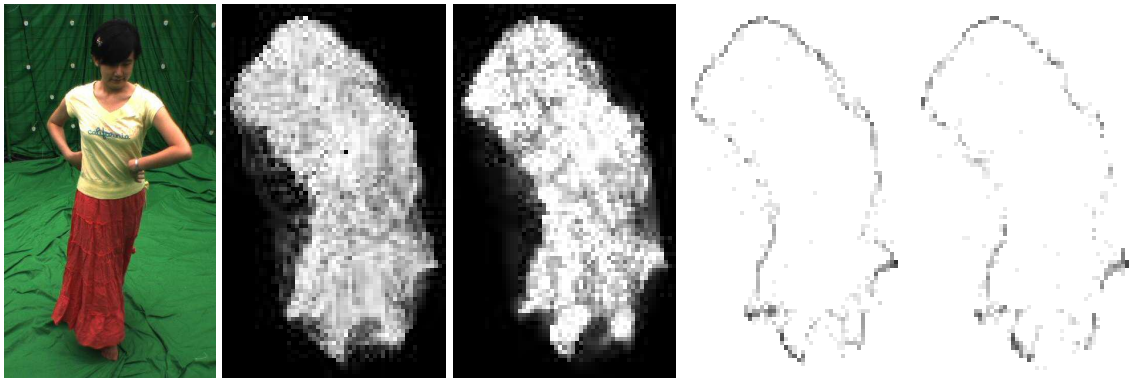
A limiting factor of our method is memory requirement. Overall, the method needs $8|V||T| \cdot 4$ bytes, one volume for the data term and photoconsistency each, two for the primal and four volumes for the dual variable. The second primal variable is needed because of the over-relaxation step in Equation (11.13). In practice memory resources are limited and smoothing over too many frames is usually not meaningful in dynamic scenes. Therefore, we limit $|T|$ to a fixed number of frames and process longer sequences with a sliding window approach for which we take the center frame of the window as the smooth solution.

11.5. Results

We applied our algorithm to several data sets provided by the INRIA 4D repository [121] and the free viewpoint video data sets from Tsinghua University provided by Liu et al. [155]. Both data sets also provide silhouette information which is quite useful in a sparse camera setup. We used the silhouette information provided with the data sets to speed up photoconsistency matching and optimization by restricting all computations to the interior of the visual hull (see Definition 10.1). Note that we do not need exact visual hull information, which is often difficult to obtain in a fully automatic manner. Due to the search space restriction, we only assume that the visual hull fully contains all objects to be recovered. Hence, the visual hull can be larger, but should not be smaller than the exact one. In some frames the silhouettes are incorrect and lead to missing scene parts in some experiments. All experiments have been computed on a Intel Xeon E5520 PC with 12GB RAM, equipped with an NVidia Tesla C2070 card and running a recent Linux distribution.

Given the relaxed solution of the energy in Equation (11.1), we extracted an isosurface at $u = 0.5$ with the Marching Cubes algorithm [156] at every time step. To better see the jittering reduction, all experiments show pure results of our algorithm after Marching Cubes without any mesh smoothing, filtering or remeshing. The following section details the photoconsistency and data term computation to explain differences and compare to previous work.

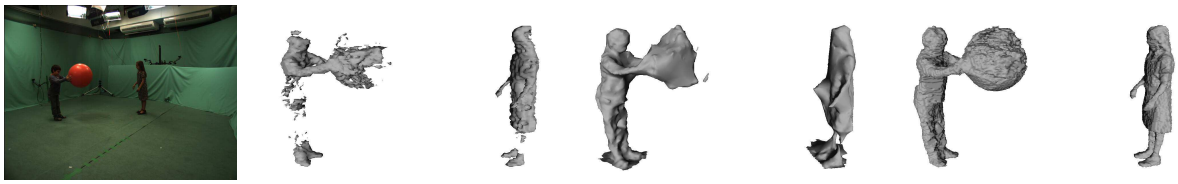
11.5.1. Photoconsistency and Data Term Evaluation



input (a) (b) (c) (d)

Figure 11.4.: Comparison of the data term from Kolev et al. [135] (a) and the proposed one (b) for a lower cross section of the skirt. Shown are the voxels' probabilities of being inside (white) and outside (black) the surface. Corresponding photoconsistencies are respectively displayed in (c) and (d). Dark pixels represent higher matching scores. Although the photoconsistency is slightly worse, the proposed data term yields sharper contours and better carves out concavities because only front facing cameras determine their shape, rather than all cameras. The volume resolution was 128x256x192.

As explained in Section 11.2.2 the data term is built based on the photoconsistency measure ρ. The quality of this measure directly influences the quality of the data term. Kolev et al. [135] iteratively improved the quality of the photoconsistency by calculating the NCC scores based on a surface normal estimate which they first take from the visual hull and later update with the solution of the surface reconstruction in an iterative manner. In the photoconsistency voting scheme as described in [135] each point $\boldsymbol{x}$ defines a ray to each camera. Point $\boldsymbol{x}$ only gets a vote if the normal corresponding to $\boldsymbol{x}$ maximizes the NCC along the whole ray in point $\boldsymbol{x}$. This means that for every point $\boldsymbol{x}$ the photoconsistency has to be calculated for all points on the corresponding camera rays with respect to the same normal. This makes the



1 of 16 input images | Kolev et al. [135] | Jancosek and Pajdla [122] | proposed

Figure 11.5.: Comparison of the reconstruction results using the data term by Kolev et al. [135] and the proposed one. Further we show the result of the method by Jancosek and Pajdla [122]. The ball has low texture information and further exhibits strong reflections which makes it difficult to reconstruct.

photoconsistency estimation inherently slow and explains the long computation times (up to 10 hours for one scene) reported in [135]. In our 4D setup we dropped this dependency by maximizing the photoconsistencies along rays independent of the normal direction. This way the photoconsistency calculations can be done independently and thus easily be parallelized to speed up computations. We simply use the viewing direction of the reference camera towards $\boldsymbol{x}$ as the surface normal estimate. As result, we achieved speedups of one or several orders of magnitude (depending on the volume resolution) for obtaining comparable results. We compared the results with our reimplementation of the normal dependent maximization and found fairly similar results. Figure 11.4 shows exemplarily results for these different photoconsistency estimation schemes.

On the left part of Figure 11.4 we compare the proposed data term with the one in [135]. We briefly repeat its definition to clarify the differences. They also define a quality measure for each camera ray defined by point $\boldsymbol{x}$ and camera j:

$$\rho_{int}^{j}(\boldsymbol{x}) = H(d_i^{\max} - d) \cdot (1 - f(\bar{C}_i(\boldsymbol{x}, d))) + (1 - H(d_i^{\max} - d)) \cdot f(\bar{C}_i(\boldsymbol{x}, d)) \tag{11.15}$$

with

$$H(x) = \mathbf{1}_{\{x<0\}} = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{11.16}$$

being the Heavyside step function switching between two different costs depending on whether d is larger or smaller than $d_i^{\max}$, i.e. if the point $\boldsymbol{x}$ is either before or behind the voting location. The data term is then defined as the average of $\rho_{int}^{j}(\cdot)$ over all cameras. The key difference to our approach is that their model influences the data term in front of *and* behind the camera vote while our approach only influences the data term between the camera and the camera vote. This global influence in their model degrades the quality of back faces and other object parts which are unrelated to the camera vote. This is visible in Figure 11.4 showing the differences in the data term, as well as in Figure 11.5 which depicts a resulting surface reconstruction. For comparison we also show the reconstruction result of Jancosek and Pajdla [122]. The scenes with the gymnastic ball are especially challenging because the ball surface has low texture information and a shiny surface. In Figure 11.6 we compared the output of our method with the methods by Jancosek and Pajdla [122] and to the ones of Liu et al. [155] who provided the data. Both methods yield much smoother surface reconstructions, but also blur fine scale details like the hand. Figure 11.7 shows more reconstruction results of our methods on various data sets. Table 11.1 lists average computation times for the experiments depicted in Figure 11.7.

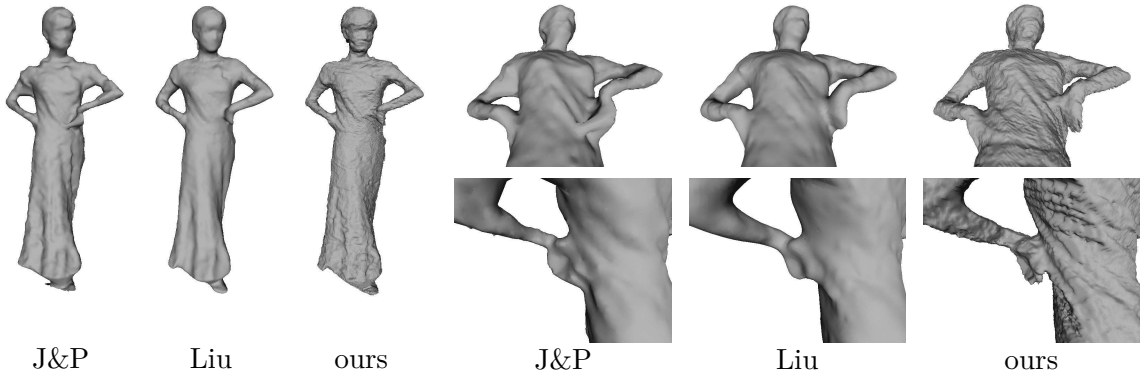


Figure 11.6.: Comparison of the proposed method for $|T| = 1$ with other 3D reconstruction methods. From left to right (twice): J&P=Jancosek and Pajdla [122], Liu=Liu et al. [155] and the proposed method. The approach by Jancosek and Pajdla wrongly connects points at the hand and the armpits. Our approach method better preserves several details such as the hand.

data set	volume size	pc+d	opt
kick one	384^3	89	28/93/-
boy cartwheel	$384 \times 384 \times 256$	21	18/59/-
children playing	$384 \times 384 \times 256$	18	18/60/-
adult child	384^3	43	31/91/-
red skirt	256^3	90	10/31/88

Table 11.1.: Average runtimes per frame for our method on different data sets for the photoconsistency and data term estimation (pc+d) and the surface optimization (opt) for different sizes of $|T| \in \{1, 3, 5\}$. Timings are in seconds/frame for different temporal window sizes. In comparison the method by Jancosek and Pajdla [122] computed 600-1200 seconds/frame.

11.5.2. Temporal Regularization

We studied the influence of the temporal window size $|T|$ and weighting $g_t = \exp(-a|\nabla_t f|^b)$ in Equation (11.2). Figure 11.9 gives an overview for $|T| \in \{3, 5, 7\}$ (horizontal) and different $a \in \{0.001, 1\}$ (left, vertical). The effect of g_t on the solution is mainly governed by parameter a. When a approaches zero the temporal regularization gets maximal and the reconstructed surface tends towards the intersection with neighboring time slices (see the disappearance of the lower leg part in Figure 11.9, top row). An illustration of the exponential weighting via parameter a is given in Figure 11.8. We did not observe significant differences for varying values of b and set $b = 1$ in all experiments. The differences are largest between window sizes $|T| = 1$ and $|T| = 3$. Choosing larger window sizes only led to subtle differences which do not pay off the increase in computation time and memory resources. Since no other 4D reconstruction implementations are publicly available and it is difficult to obtain ground truth geometry, we visually compared our method with (a) time-independent reconstruction by Jancosek and Pajdla [122], (b) time-independent reconstruction as proposed with $|T| = 1$, (c) temporal Gaussian smoothing of (b) as post processing for temporal smoothness, and (d) the proposed method with $|T| = 3$. In particular, we computed a smoothed occupancy labeling $\bar{u}$ from the time-independent result

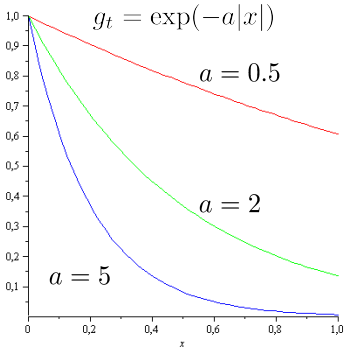


Figure 11.8.: Illustration of the exponential temporal weighting $g_t = \exp(-a|\nabla_t f|^b)$. The effect of parameter a is plotted for $b = 1$ and $x = \nabla_t f$. This weighting suppresses temporal smoothing in the presence of fast motion (i.e. large $|\nabla_t f|$).

Figure 11.7.: Results of our framework on several data sets for $|T| = 3$. For the cartwheel sequence we selected frame numbers (120,130,222,347,442) and for red skirt frame numbers (41,45,50,55,58). Please refer to the supplementary material [7] for video sequences.

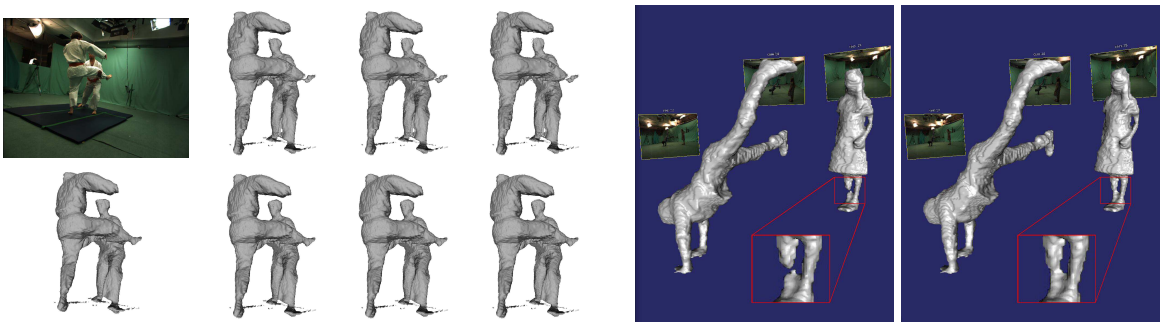
$\hat{u}$ as follows:

$$\bar{u}(\boldsymbol{x}, t) = \frac{1}{Z} \sum_{i=0}^{|T|-1} \exp\left[-\frac{(i - |T|/2)^2}{2\sigma^2}\right] \hat{u}(\boldsymbol{x}, t + i - |T|/2) \tag{11.17}$$

Figure 11.10 shows a representative frame for each method. Generally, the Gaussian filtering cannot reach the same level of smoothness as (d) while preserving fast moving object parts. For preserving fast movements σ needs to be chosen very small such that voxel jittering is barely reduced. The proposed method balances these issues much better.

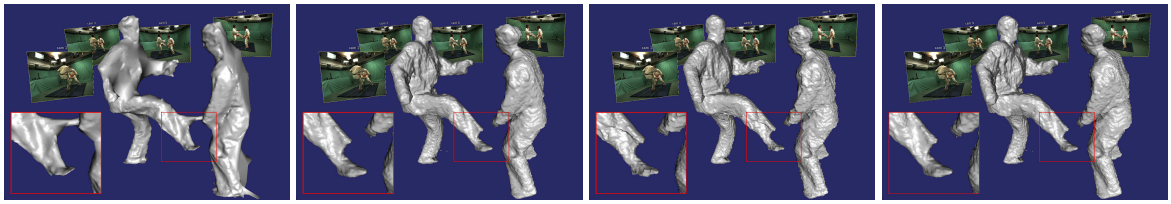
11.6. Conclusion

In this chapter we presented a novel approach to space time multi-view 3D reconstruction that generalizes several previous works into a 4D setting. In order to get competitive reconstructions on wide-baseline camera setups we further proposed a novel data term that better preserves concavities and fine details. 3D reconstruction results compare favorably to other works. Our approach directly accounts for temporal surface coherence within the reconstruction process. In comparison to single frame-by-frame reconstruction our approach clearly reduces the amount of noise on the estimated surface. In several experiments we



s$|T| = 1$ $|T| = 3$ $|T| = 5$ $|T| = 7$ $|T| = 1$ $|T| = 3, a = 0.5$

Figure 11.9.: Effect of the temporal regularization. The approach allows to impose temporal regularity over multiple time steps $|T|$. For a small weight of temporal smoothness ($a = 1$, left bottom row) the regularity reduces the jittering of voxels over time (see supplementary video [7]), whereas for strong temporal smoothness ($a = 0.001$, left top row) the regularization starts to deteriorate fast moving structures like the right foot. Temporal coherence also improves reconstructions with weak photoconsistencies in single time frames (right).



(a) Jancosek and Pajdla [122] ($|T| = 1$) (b) time-independent ($|T| = 1$) (c) temporal filtering ($|T| = 3$) (d) proposed ($|T| = 3$)

Figure 11.10.: Comparison of different reconstruction techniques. (a) produces strong surface jittering, wrongly connects the leg and hand and misses parts of the head. (b) Voxel jittering is visible. (c) Voxel jittering can be reduced, but fast moving object parts start disappearing, e.g. the foot. The edge on the lower leg is an artifact of the averaging of consecutive time frames. (d) Due to the weighting and the TV-regularization the problems of (c) can be balanced much better (see also the supplementary video [7]).

showed the viability of the proposed framework. To our knowledge, this is the first time that space-time 3D reconstruction was formulated as a convex variational problem. The solutions are provably optimal in terms of the objective function, independent of initialization and recover fine details.

12. Surface Normal Integration and Spatially Anisotropic Regularization

If you do not change direction, you may end up where you are heading.

Laozi
(Philosopher and Poet, 604 BC - 531 BC)

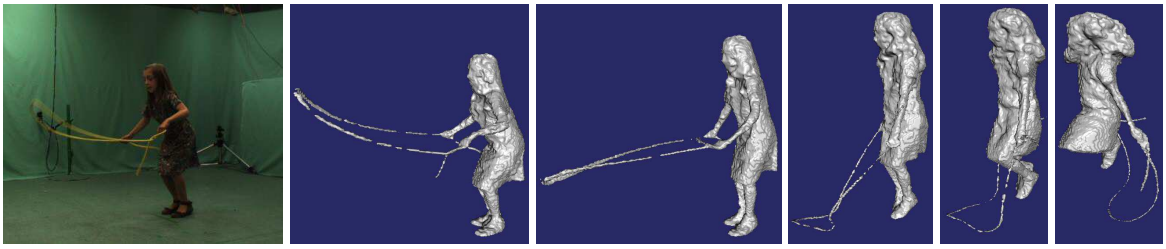


Figure 12.1.: Frame 17 (of 409) from the "jumping rope sequence" [121] and corresponding reconstructions of this and the following time frames computed with the proposed method. By minimizing a single convex functional, we obtain a family of reconstructions over time. By integrating normal information into the photoconsistency estimation and into an anisotropic space-time regularization, we are able to preserve fine scale details such as the (substantially motion-blurred) rope.

In this chapter we show that surface normal information allows to significantly improve the accuracy of a spatio-temporal multi-view reconstruction. On one hand, normal information can improve the quality of photometric matching scores. On the other hand, the same normal information can be employed to drive an adaptive anisotropic surface regularization process which better preserves fine details and elongated structures than its isotropic counterpart. We demonstrate how normal information can be used and estimated and explain crucial steps for an efficient implementation. Experiments on several challenging multi-view video data sets show clear improvements over state-of-the-art methods. This chapter has been published in [8].

12.1. Introduction

The extension of multi-view 3D reconstruction approaches to the spatio-temporal domain is far from straightforward: Firstly, with the processing of huge amounts of data, computational speed becomes more important. Algorithms which take around an hour for a single reconstruction are hardly scalable to multi-view videos taken at 30 frames per second. Secondly, integrating temporal regularization gives rise to a substantial increase in memory requirements because the reconstructions for multiple time steps need to be computed jointly. Thirdly, the acquisition of actions over time brings about substantial motion blur of fast moving structures – see the rope in Figure 12.1. And lastly, one typically uses far fewer cameras with lower resolution (the synchronization and joint acquisition being tedious) such that classical photoconsistency approaches often break down.

12.1.1. Related Work

In this section we give a brief overview of most related related works on multi-view stereo reconstruction which used or estimated surface normals in their approach.

An early work considering surface normals while estimating a 3D surface is by Zabulis and Daniilidis [247]. They estimate voxel occupancy at the surface and corresponding normals in a voxel grid by locally maximizing corresponding surface patch correlation values. The optimization is spatially local and results in rather noisy and disconnected surfaces. Furukawa et al. [91] proposed to jointly estimate depth and orientation of surface patches by means of an oriented point cloud which can then be transformed into a mesh e.g. via Poisson surface reconstruction [129]. Goesele et al. [93] built a system for reconstructing 3D scenes from internet photo collections. They show that optimizing surface normal information with respect to the photoconsistency measure significantly improves the reconstruction quality. Both methods [91, 93] are based on an oriented point cloud which is grown and filtered iteratively around existing matches by starting from sparse feature matches. Generally, an extension of such models into a spatio-temporal domain is by no means straightforward because the correspondence of points over time needs to be identified first.

Ladikos et al. [145] used a narrow band graph-cut approach for multi-view reconstruction. They jointly maximized a normalized cross-correlation (NCC) photoconsistency measure and computed the best normal by discretely sampling a dense set of normals in the cone around an initial normal estimate. Vu et al. [234] proposed a reconstruction pipeline for large scale scenes. They also experimented with choosing different orientations during the reconstruction and keeping only the best one, yet they did not observe noticeable improvements. Kolev et al. [136] improved the results of multi-view 3D reconstruction with an anisotropic regularizer and a given normal field. We use a similar regularizer, but additionally discuss how to compute such a normal field and how it can be used to improve photometric measures.

Wu et al. [242] estimate normals from multi-view video based on a coarse tracked shape model over consecutive time steps and use this information to augment the same model with the estimated fine details.

There exist a wide range of works which aim to reconstruct surfaces from a given normal vector field. Note that these works aim to solve a different problem to the one we consider in this chapter. Nevertheless, some of these works share similar ideas and approaches and are worth mentioning in this context. Chang et al. [50] integrate several normal fields captured by structured light techniques from multiple views and merge all information within a level-set framework to recover the full 3D shape of a target object. Vlasic et al. [230] use a costly camera dome setup with structured light to accurately estimate normal and depth maps and merge them so single model in a data-driven manner. Weinmann et al. [237] use structured light to estimate the normals and surface location of objects with mirroring surface properties. Kazhdan et al. [129, 130] wrote a series of works on "Poisson surface reconstruction" from oriented point clouds which gained a lot of popularity in the community as a final 3D reconstruction step to obtain hole-free "watertight" meshes. Similar to our approach they compute a surface via a binary interior/exterior indicator function. The sparse normal information is integrated by solving a Poisson equation on an octree-based data-adaptive grid.

12.1.2. Contributions

In this chapter, we propose a convex variational approach to space-time reconstruction which estimates surface normal information and integrates it into the photoconsistency estimation as well as into an anisotropic spatio-temporal total variation regularization. As such, the

proposed method generalizes the works of Chapter 11 ([7]) and the one of Kolev et al. [136] on anisotropic regularization. Although [136] already studied anisotropic regularization they did not estimate normals but used the normals from [91]. The combination of these methods, [136] and [91], is more than 40 times slower than our method as [136] alone needs about 1h to compute a single frame. In contrast, our method only takes about 3 minutes per frame including normal estimation and temporal regularization due to the proposed efficient implementation. Moreover, the method by Kolev et al. [136] does not work well on the 4D data sets we consider, as shown in Figure 11.5 of the previous Chapter 11. With the estimated normals at hand, we further propose an improvement of the photoconsistency voting scheme by Hernández and Schmitt [77] resulting in superior accuracy especially for sparse camera setups.

12.2. Variational Space-Time Reconstruction Model

Similar to the previous Chapter 11, we aim to find a smooth hypersurface Σ in the spatio-temporal space $V \times T$ in which $V \subset \mathbb{R}^3$ represents the spatial and $T \subset \mathbb{R}_{\geq 0}$ the temporal domain. A non-static scene is observed from N cameras with known projections $\{\pi_i\}_{i=1}^N$ and approximate silhouettes $\{S_i(t)\}_{i=1}^N$. Similar to Chapter 11, we assume the silhouettes to fully enclose the object of interest and restrict the solution space to the visual hull (Definition 10.1). We do not rely on exact silhouettes as they are difficult to estimate in a general 4D setup. Hence, methods using exact silhouettes such as [62] are not applicable. Again, we will drop temporal indices whenever possible for better readability.

First, we introduce a binary labeling function $u : V \times T \to \{0, 1\}$ to represent the hypersurface Σ by means of an inside-outside labeling in each point. This implicit surface representation easily deals with topology changes and allows to compute minimal surfaces that align with locations of high photometric consistency in a globally optimal manner [135]. We compute a hypersurface as a minimum of the following energy.

$$E(u) = \int_{V \times T} \Big[\, |\nabla_{\boldsymbol{x}} u|_{D_{\boldsymbol{x}}} + g_t |\nabla_t u| + \lambda \, f u \, \Big] \; d\boldsymbol{x} dt \tag{12.1}$$

The parameter $\lambda \geq 0$ steers the smoothness of the solution by balancing the costs of the regularization term and the data term. The function $f : V \times T \to \mathbb{R}$ represents unary potentials which indicate local preferences for either an interior or an exterior label based on the photoconsistency being defined in the next section. The task of the regularization term is to reject outliers, to deal with locations of missing data and to favor a spatially and temporally smooth surface. The regularization term consists of two terms, one for the anisotropic spatial regularization with the norm defined as $|\boldsymbol{y}|_{D_{\boldsymbol{x}}} = \langle \boldsymbol{y}, D_{\boldsymbol{x}} \boldsymbol{y} \rangle^{1/2}$ (see [168] for details) and the other term takes care of the temporal regularization. Both terms are detailed in the following.

Spatial Regularization. The symmetric positive-definite matrix $D_{\boldsymbol{x}}$ accounts for an anisotropic spatial regularization and is defined similarly as in [185].

$$D_{\boldsymbol{x}}(\boldsymbol{x}, t) = \rho(\boldsymbol{x}, t)^2 \boldsymbol{n}\boldsymbol{n}^T + \boldsymbol{n}_0 \boldsymbol{n}_0^T + \boldsymbol{n}_1 \boldsymbol{n}_1^T \; . \tag{12.2}$$

It lowers smoothing in the direction of the surface normal $\boldsymbol{n} \in \mathbb{R}^3$ and favors smoothness along the corresponding tangential directions $\boldsymbol{n}_0$ and $\boldsymbol{n}_1 = \boldsymbol{n} \times \boldsymbol{n}_0$. The *anisotropic* total variation norm $|\nabla_{\boldsymbol{x}} u|_{D_{\boldsymbol{x}}}$ is a generalization of the total variation norm [168], because for $D_{\boldsymbol{x}}(\boldsymbol{x}, t) = \operatorname{diag}\left(\rho(\boldsymbol{x}, t)^2\right)$ the regularization term reduces to the *isotropic* weighted total variation norm $\int \rho \, |\nabla_{\boldsymbol{x}} u| \; d\boldsymbol{x}$.

The photoconsistency measure $\rho : V \times T \to [0, 1]$ is detailed in the next section. Essentially, $D_{\boldsymbol{x}}$ performs a change of basis and aligns the local coordinate system along the favored surface normal $\boldsymbol{n}$. The vector components in normal direction are downscaled with the photoconsistency measure ρ while the tangential directions remain untouched. As a result, gradients in normal direction are less penalized then other directions and ∇u is more likely to be aligned to $\boldsymbol{n}$. On the one hand the anisotropic regularization better preserves small scale surface details [136], on the other hand it is important when reconstructing fine elongated structures [185] like human arms, or parts of clothes and hair.

In contrast to the *isotropic* weighted total variation model ($D_{\boldsymbol{x}} = \rho^2 \mathbb{I}_{3\times3}$) only *one* spatial direction is downscaled by $\rho(\boldsymbol{x})$, while in the isotropic case *all* spatial directions are weighted. Consequently, the overall diffusivity is larger in the anisotropic case and therefore the smoothness parameter λ in Equation (12.1) has to be chosen larger in order to obtain results of similar smoothness. Since we will estimate normals based on photoconsistency information this weighting makes perfect sense, because a low photoconsistency measure also indicates that the corresponding normal estimate is uncertain.

Temporal Regularization. In Equation (12.1) function $g_t : V \times T \to \mathbb{R}_{\geq 0}$ regulates the temporal smoothness. By setting $g_t(\boldsymbol{x}, t) = \exp\left(-a|\nabla f(\boldsymbol{x}, t)|\right)$ we make it dependent on the data term in order to reduce temporal smoothing in regions with fast motion. The purpose of this regularization is to reduce surface jittering in scene parts with slow motion. The effect of this term is studied in Chapter 11. We used values for a between 0.2 and 1 in our setting.

12.3. Surface Normal Integration

Normal information is used in all stages of our approach, namely during the photoconsistency and data term estimation as well as during the global surface optimization.

12.3.1. Photoconsistency and Data Term Estimation

In order to estimate photometric consistency measures and to build a corresponding data term, we use the same model as proposed in Section 11.2.1 in the previous chapter, but integrate available normal information to obtain better matching scores. For every time step we estimate the photometric consistency of a point on the surface by means of a cost function $C_i : V \times \mathbb{R} \to \mathbb{R}$ based on the NCC score of corresponding small image patches surrounding the projection of that point in each camera.

$$C_i(\boldsymbol{x}, d) = \sum_{j \in \mathcal{C}' \setminus i} w_i^j(\boldsymbol{x}) \cdot \mathrm{NCC}\Big(\pi_i\big(r_i(\boldsymbol{x}, d)\big), \pi_j\big(r_i(\boldsymbol{x}, d)\big)\Big) \tag{12.3}$$

The value d is the Euclidean distance of $\boldsymbol{x}$ from camera center i along camera ray $r_i(\boldsymbol{x}, \cdot)$ through point $\boldsymbol{x}$. $\mathcal{C}' \subset \mathcal{C}$ is a subset of front-facing cameras of which the angle between the viewing directions is below γ_{max}=85°. The contribution of each camera is weighted by a normalized Gaussian weight $w_i^j(\boldsymbol{x})$ of the angle between the voxel-to-camera directions of cameras i and j. Furthermore, we discard unreliable correlation values by setting $C_i(\cdot)$ to zero if it falls below a threshold $\tau_{\mathrm{ncc}} = 0.3$. To account for image distortion between two cameras during the NCC computation the image coordinates are mapped with the homography $H_{ij} = (\boldsymbol{n}^T\boldsymbol{x})R_{ij}^T - R_{ij}^T\boldsymbol{t}_{ij}\boldsymbol{n}^T$, with $\boldsymbol{n}$ being the surface normal, $\boldsymbol{x} \in V$ the current point and $R_{ij} \in SO(3)$, $\boldsymbol{t}_{ij} \in \mathbb{R}^3$ being the relative rotation and translation between local coordinates of cameras i and j [79].

Since the correlation scores $C_i(\cdot)$ are usually noisy and contain many local maxima we denoise them with the voting scheme by Hernández and Schmitt [77] and define the photoconsistency measure $\rho(\boldsymbol{x})$ for the regularizer as

$$\rho(\boldsymbol{x}) = \exp\Big[-\mu \sum_{i \in \mathcal{C}'} \underbrace{\delta\big(d_i^{\max} = \text{depth}_i(\boldsymbol{x})\big) \cdot C_i(\boldsymbol{x}, d_i^{\max})}_{\text{VOTE}_i(\boldsymbol{x})} \Big] \ . \tag{12.4}$$

This scheme accumulates only the best score along each camera ray. The point with maximum score is expressed by its distance to the camera center

$$d_i^{\max} = \arg\max_d C_i(\boldsymbol{x}, d) \ . \tag{12.5}$$

In comparison, for most 3D reconstruction approaches that first estimate depth maps before fusing them into a single 3D model, e.g. [248], the matching scores of single depth estimates are not considered in the depth fusion process. In contrast, the voting scheme accumulates matching scores and we hand them over to the global surface estimation. Another significant difference to such methods is the missing regularization of depth values in the image domain, which often helps to avoid depth ambiguities and to suppress noise. We therefore propose to introduce a dependency between neighboring camera rays by the following modification of the voting scheme in Equation (12.5):

$$d_i^{\max} = \arg\max_d \int_{V_{\boldsymbol{x}}} C_i(\boldsymbol{x} - \boldsymbol{y}, d)\ \mathcal{G}(\boldsymbol{y}; \Sigma_{\boldsymbol{n}})\ d\boldsymbol{y} \ , \tag{12.6}$$

where $V_{\boldsymbol{x}} \subset V$ is a small volume surrounding $\boldsymbol{x}$. Each value of $C_i(\cdot)$ represents the matching score of a small surface patch with location $\boldsymbol{x}$ and orientation $\boldsymbol{n}$ and should also influence neighboring matching scores according to the patch size. We model this dependency by a Gaussian convolution of the matching scores before the maximization. Again, the normal information comes in handy to better represent the shape of the surface patch by an anisotropic 3D Gaussian $\mathcal{G}(\cdot)$ with covariance matrix $\Sigma_{\boldsymbol{n}} = R_{\boldsymbol{n}} \,\text{diag}(\sigma_n^2, \sigma_t^2, \sigma_t^2) R_{\boldsymbol{n}}^T$. σ_n and σ_t are the standard deviations for normal and tangential directions and rotation matrix $R_{\boldsymbol{n}}$ aligns the x-axis of the coordinate system with the normal $\boldsymbol{n}$. This scheme effectively denoises depth hypotheses and improves the quality of matching scores for piecewise smooth surfaces as it helps to avoid local maxima by integrating information from neighboring viewing rays.

In order to avoid trivial solutions of energy (12.1) the photoconsistency is further imposed by means of an unary data term f, defined as the log-probability ratio

$$f(\boldsymbol{x}, t) = -\ln\left(\frac{1 - P(\boldsymbol{x} \in int(\Sigma))}{P(\boldsymbol{x} \in int(\Sigma))}\right) \ . \tag{12.7}$$

The probability $P(\boldsymbol{x} \in int(\Sigma))$ that point $\boldsymbol{x}$ belongs to the interior of surface S is defined based on the voting locations and qualities of corresponding camera rays $r_i(\boldsymbol{x}, \cdot)$ through point $\boldsymbol{x}$

$$P(\boldsymbol{x} \in int(\Sigma)) = \prod_{i=1}^{N} \prod_{j=1}^{N} \prod_{\text{depth}_i(\boldsymbol{x}) < d \leq d_i^{\max}} \frac{1}{Z_j} \exp\Big[-\eta \cdot \text{VOTE}_j(r_i(\boldsymbol{x}, d))\Big] \tag{12.8}$$

Z_j is a normalization constant and parameter η steers how many cameras and which matching scores are necessary to favor an exterior label for all points from $\boldsymbol{x}$ towards the camera. Intuitively, the data term represents a probabilistic space carving and due to the restriction of the solution space to the visual hull, the visual hull is the fall back solution for all areas where photometric information is insufficient.

12.3.2. Normal Estimation

Similar to [145] we experimented with estimating the normal direction by global maximization of the NCC score via discrete sampling around the camera-to-point direction. Generally, pointwise optimization of the surface normal is prone to local minima and we merely got noisy and unsatisfactory results with this approach. Similar results have also been reported by [234]. We also tried estimating normals based on the data term f as done in [185] which also yields defective normals due to the fact that f is very noisy and misses a lot of data for most of our experiments. Kolev et al. [135] estimated normal directions for the photoconsistency computation based on the visual hull. Especially in sparse camera setups we found that the visual hull does not provide good normal estimates for recovering concavities.

Instead we use the camera-to-point direction as a first normal estimate for photoconsistency estimation which is a common (inherent) assumption in most stereo-based methods. We then compute a surface with isotropic spatial regularization and use the surface normals of this solution for a second pass of photoconsistency, data term estimation and surface optimization with anisotropic spatial regularization. For that purpose the surface normals are propagated in space by means of a signed distance function (Section 12.5). In sum, we make use of surface normals at three places within our method: (a) NCC score, (b) voting scheme regularization and (c) anisotropic surface regularization. We run our algorithm in two passes:

Pass 1: camera-to-point direction as normal for (a) and (b), isotropic surface regularization with high λ for (c)

Pass 2: normals from the previous pass for (a),(b) and (c) with lower λ for surface smoothness as desired

This scheme could be further iterated, but in our experience two passes achieve the best trade-off between quality improvements and additional computation time.

12.4. Optimization

In order to deal with the non-differentiability of the total variation norm by using the Legendre-Fenchel transform we first rewrite the anisotropic spatial regularization term in energy (12.1) based on the following equalities.

$$\begin{aligned} |\nabla_{\boldsymbol{x}} u|_{D_{\boldsymbol{x}}} &= \sqrt{(\nabla_{\boldsymbol{x}} u)^T D_{\boldsymbol{x}} \nabla_{\boldsymbol{x}} u} \\ &= \sqrt{(\nabla_{\boldsymbol{x}} u)^T \left(D_{\boldsymbol{x}}^{1/2^T} D_{\boldsymbol{x}}^{1/2} \right) \nabla_{\boldsymbol{x}} u} \\ &= \sqrt{\left(D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} u \right)^T \left(D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} u \right)} \\ &= \left| D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} u \right|_2 \end{aligned} \tag{12.9}$$

Since matrix $D_{\boldsymbol{x}}$ is positive-definite, it has a unique square root which can be found by diagonalization:

$$\begin{aligned} {D_{\boldsymbol{x}}}^{1/2} &= \left(\rho(\boldsymbol{x},t)^2\, \boldsymbol{n}\boldsymbol{n}^T + \boldsymbol{n}_0\boldsymbol{n}_0^T + \boldsymbol{n}_1\boldsymbol{n}_1^T\right)^{1/2} \\ &= \left(\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^{-1}\right)^{1/2} \\ &= \mathbf{P}\boldsymbol{\Lambda}^{1/2}\mathbf{P}^{-1} \\ &= \rho(\boldsymbol{x},t)\, \boldsymbol{n}\boldsymbol{n}^T + \boldsymbol{n}_0\boldsymbol{n}_0^T + \boldsymbol{n}_1\boldsymbol{n}_1^T \end{aligned} \tag{12.10}$$

$$\text{with} \quad \mathbf{P} = \begin{bmatrix} | & | & | \\ \boldsymbol{n} & \boldsymbol{n}_0 & \boldsymbol{n}_1 \\ | & | & | \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Lambda} = \begin{bmatrix} \rho(\boldsymbol{x},t)^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{12.11}$$

Since matrix $\mathbf{P}$ is composed of three orthogonal vectors as its columns, we have $\mathbf{P}^{-1} = \mathbf{P}^T$ and computing the square root of $D_{\boldsymbol{x}}$ is as simple as computing $D_{\boldsymbol{x}}$ itself.

The minimization problem in Equation (12.1) becomes convex by relaxing the image of function u to $[0,1]$. We globally minimize the energy with the preconditioned primal-dual algorithm by Pock and Chambolle [175] which solves certain saddle-point problems efficiently. To this end, we introduce a dual variable $\boldsymbol{p} : V{\times}T \to \mathbb{R}^4$ which tackles the non-differentiability of the total variation norm:

$$u^* = \arg\min_u E(u) \tag{12.12}$$

$$= \arg\min_u \int_{V\times T} \left[\, \left| D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} u \right|_2 + g_t |\nabla_t u|_2 + \lambda\, f u \,\right] d\boldsymbol{x}dt \tag{12.13}$$

$$= \arg\min_u \max_{\boldsymbol{p}\in P} \int_{V\times T} \left[\langle \boldsymbol{p}_{\boldsymbol{x}}, D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} u \rangle + \langle \boldsymbol{p}_t, \nabla_t u \rangle + \lambda\, f u \right] d\boldsymbol{x}dt \ , \tag{12.14}$$

with set P being defined below. The algorithm converges to the globally optimal solution by iterating a projected gradient descent and gradient ascent for the primal and dual variables respectively. The pointwise update equations for each iteration k are as follows.

$$\boldsymbol{p}^{k+1} = \Pi_P \left[\boldsymbol{p}^k + \sigma \left(D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} \bar{u}^k \ , \ \nabla_t \bar{u}^k \right)^T \right] \tag{12.15}$$

$$u^{k+1} = \Pi_{[0,1]} \left[u^k + \tau \Big(\operatorname{div} \left((D_{\boldsymbol{x}}^{1/2} \boldsymbol{p}_{\boldsymbol{x}}^{k+1} \ , \ \boldsymbol{p}_t^{k+1})^T \right) - \lambda f \Big) \right] \tag{12.16}$$

$$\bar{u}^{k+1} = 2u^{k+1} - u^k \tag{12.17}$$

$\Pi_{[0,1]}$ projects u onto the unit interval $[0,1]$ via simple thresholding. The projection onto the set $P = \{\boldsymbol{p} = (\boldsymbol{p}_{\boldsymbol{x}}, \boldsymbol{p}_t)^T : V{\times}T \to \mathbb{R}^4 | \ \|\boldsymbol{p}_{\boldsymbol{x}}\| \leq 1, |\boldsymbol{p}_t| \leq 1\}$ can be done as follows:

$$\Pi_P(\boldsymbol{p}) = \left(\frac{\boldsymbol{p}_{\boldsymbol{x}}}{\max(1, \|\boldsymbol{p}_{\boldsymbol{x}}\|)}, \max\left(-1, \min(1, \boldsymbol{p}_t)\right) \right)^T \tag{12.18}$$

Set P can be imagined like a "capsule pill", i.e. a 3D ball shifted along the 4th dimension. The step sizes σ and τ are chosen adaptively by keeping track of the corresponding operator norms as suggested in [175]. Note that the linear operators that transform between primal and dual space contain the discretized differential operators and the diffusion matrix $D_{\boldsymbol{x}}$ which need to be considered for the preconditioning.For the primal variable u we impose Neumann boundary conditions for both spatial and temporal derivatives and accordingly

Dirichlet boundary conditions for $\boldsymbol{p}$:

$$\nabla_{\boldsymbol{n}} u\Big|_{\partial(V\times T)} = \langle \nabla u, \boldsymbol{n} \rangle \Big|_{\partial(V\times T)} = 0 \qquad \text{and} \qquad \boldsymbol{p}\Big|_{\partial(V\times T)} = 0 \ , \tag{12.19}$$

where $\boldsymbol{n}$ is the normal to the domain boundary in this case. Note that this optimization procedure solves the relaxed optimization problem. In order to get a binary occupancy labeling one can simply threshold the values of u. However, the relaxed solution provides sub-voxel accuracy when extracting an iso-surface which we do at 0.5 using the Marching Cubes algorithm [156]. To better see voxel jittering effects all experiments show the direct outcome of this algorithm without any smoothing or remeshing. In most examples one can observe particular voxel layers, which indicates that the relaxed solution of optimization problem (12.1) is close to the solution of the original binary formulation in locations where information about the surface is strong. These discretization artifacts can be tackled with better iso-surface extraction methods or simple post-smoothing.

12.5. Implementation

Both the photoconsistency estimation and the surface optimization are highly parallelizable and have been implemented on the GPU using the NVidia CUDA framework. An efficient integration of the anisotropic regularization is challenging because in every point the derivative of the spatially and temporally varying diffusion tensor $D_{\boldsymbol{x}}$ needs to be evaluated based on the normal estimate $\boldsymbol{n}$ in each point. A straightforward implementation would easily double the overall memory consumption and render the numerical problem infeasible for reasonable volume resolutions. To save memory we do *not* precompute or save the 3×3 diffusion matrix $D_{\boldsymbol{x}}$, but we recompute $D_{\boldsymbol{x}}$ and its derivative as needed and make use of its symmetry. Further, instead of saving a dense normal field for every time step, we store a signed distance function of the previous surface estimate which requires only one additional volume per frame and allows us to densely compute normal estimates as its derivative everywhere in the volume. To compute $D_{\boldsymbol{x}}^{1/2}$ via $\boldsymbol{n}, \boldsymbol{n}_0, \boldsymbol{n}_1$, we use the Gram-Schmidt orthogonalization method starting with the local normal estimate $\boldsymbol{n}$ and the unit vector that points in the direction of the smallest absolute entry of $\boldsymbol{n}$. Further, we approximate spatial and temporal derivates of $D_{\boldsymbol{x}}^{1/2}$ with forward and backward differences for the primal and dual steps, respectively, in order to ensure the adjointness of primal and dual operators. For instance, for the spatial derivates at a fixed time step t (omitted here for better readability) we used

$$D_{\boldsymbol{x}}^{1/2} \nabla_{\boldsymbol{x}} u \quad \approx \quad D_{\boldsymbol{x}}^{1/2} \cdot \begin{pmatrix} u(x+1,y,z) - u(x,y,z) \\ u(x,y+1,z) - u(x,y,z) \\ u(x,y,z+1) - u(x,y,z) \end{pmatrix} \tag{12.20}$$

and

$$
\begin{aligned}
&\operatorname{div}\left(D_{\boldsymbol{x}}^{1/2}(x,y,z)\boldsymbol{p}_{\boldsymbol{x}}(x,y,z)\right)\\
&\quad\approx\Big(d_{11}(\,x,\,y,\,z\,)\,p_1(\,x,\,y,\,z\,) \quad +d_{12}(\,x,\,y,\,z\,)\,p_2(\,x,\,y,\,z\,) \quad +d_{13}(\,x,\,y,\,z\,)\,p_3(\,x,\,y,\,z\,)\Big)-\\
&\qquad\Big(d_{11}(x-1,y,z)\,p_1(x-1,y,z) \quad +d_{12}(x-1,y,z)\,p_2(x-1,y,z) \quad +d_{13}(x-1,y,z)\,p_3(x-1,y,z)\Big)+\\
&\qquad\Big(d_{21}(\,x,\,y,\,z\,)\,p_1(\,x,\,y,\,z\,) \quad +d_{22}(\,x,\,y,\,z\,)\,p_2(\,x,\,y,\,z\,) \quad +d_{23}(\,x,\,y,\,z\,)\,p_3(\,x,\,y,\,z\,)\Big)-\\
&\qquad\Big(d_{21}(x,y-1,z)\,p_1(x,y-1,z) \quad +d_{22}(x,y-1,z)\,p_2(x,y-1,z) \quad +d_{23}(x,y-1,z)\,p_3(x,y-1,z)\Big)+\\
&\qquad\Big(d_{31}(\,x,\,y,\,z\,)\,p_1(\,x,\,y,\,z\,) \quad +d_{32}(\,x,\,y,\,z\,)\,p_2(\,x,\,y,\,z\,) \quad +d_{33}(\,x,\,y,\,z\,)\,p_3(\,x,\,y,\,z\,)\Big)-\\
&\qquad\Big(d_{31}(x,y,z-1)\,p_1(x,y,z-1) \quad +d_{32}(x,y,z-1)\,p_2(x,y,z-1) \quad +d_{33}(x,y,z-1)\,p_3(x,y,z-1)\Big)\ ,
\end{aligned}
\tag{12.21}
$$

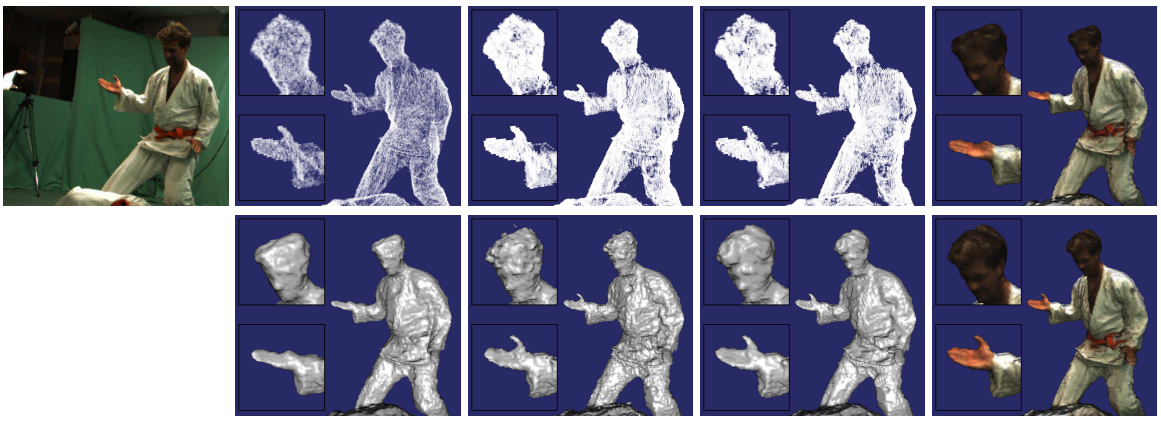
where d_{ij} denote the elements of matrix $D_{\boldsymbol{x}}^{1/2}$ and p_j denote the elements of vector $\boldsymbol{p}_{\boldsymbol{x}}$.

As a result, the total amount of required memory per frame is $9 \cdot |V{\times}T| \cdot 4$ bytes. One volume for the data term, photoconsistency and signed distance function each, two for the primal and four volumes for the dual variable. The second primal variable is needed for the over-relaxation step in Equation (12.17). Based on the experimental results of Section 11.5.2 in the previous chapter, we used $|T| = 3$ and processed longer sequences with a sliding time window approach considering also the frames before and after the current one and took the center frame of the window as the temporally smoothed solution. Further significant memory savings (factor 1/4 to 1/10) and speedups (factor 25 to 30) can be achieved by restricting all computations and data structures to the visual hull using indexed lists. For the scenes we have evaluated in this chapter this approach is very useful because the size of volume V is large to capture the dynamics of the scene over time. Thus only a small amount of the volume is labeled as interior.

12.6. Results

We tested our approach on several multi-view sequences with 16 cameras and 1624×1224 image resolution from the INRIA 4D repository [121]. We computed all experiments on a Linux-based PC with a 2.27GHz Xeon CPU, 24GB RAM and an NVidia Titan 6GB graphics card. For quality assessment we compared our method with several state-of-the-art 3D and 4D reconstruction methods: PMVS [91] + Poisson surface reconstruction [129], Jancosek and Pajdla [122] and the isotropic method in Chapter 11 (also referred to as [7] for brevity). For all methods we used default parameters, full input image resolution and provided approximate silhouettes if possible (all except [122]).

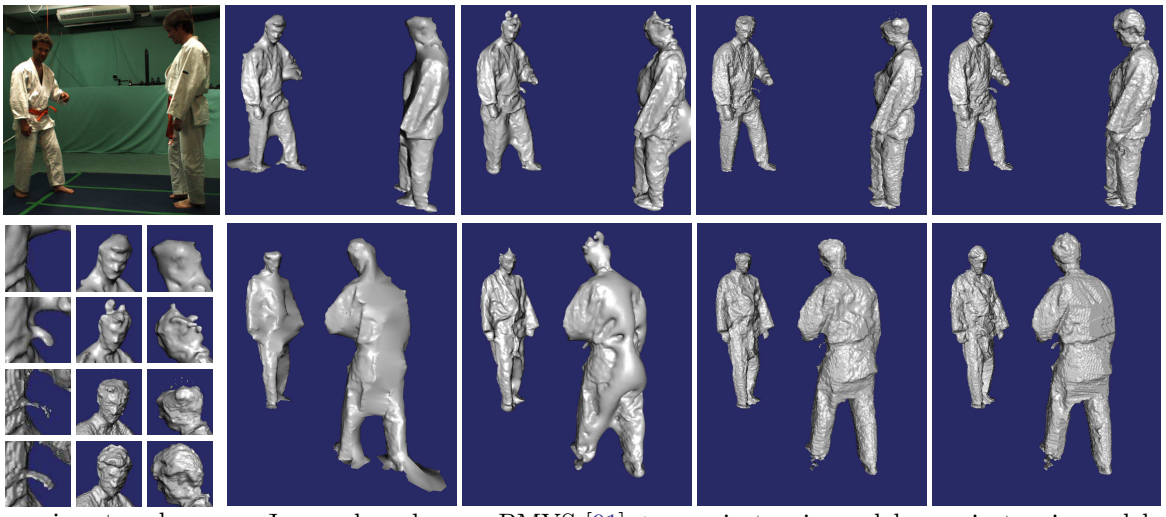
Figure 12.2 shows the influence of normal information on the reconstruction quality in every step of the reconstruction process. For the first pass of our method we used higher standard deviations ($\sigma_n = 0.4, \sigma_t = 0.9$) for the anisotropic Gaussian smoothing kernel in Equation (12.6) to achieve a higher denoising of NCC scores from potentially wrong initial normal estimates. The anisotropic smoothing of the NCC scores makes them more discriminative in comparison to the viewing ray independent voting scheme used in the previous Chapter 11 and leads to more distinctive votes (top row). Fine details are only preserved for low smoothness values (bottom row). In the second pass we reduced the Gaussian smoothing ($\sigma_n = 0.3, \sigma_t = 0.7$) to better preserve fine details in the reconstruction. The normal estimates from pass 1 further improve the photoconsistencies (e.g. the hair) and the anisotropic



input | without normal integration [7] | pass 1 | pass 2 (proposed) | comparison column 2 & 4

Figure 12.2.: Effects of the proposed normal integration. Column 2 shows the results without normal integration. The photoconsistency $\rho(x)$ (top) is noisy and less discriminative leading to a reconstruction (bottom) that misses details like the thumb and the hair due to low photometric information. In comparison the photoconsistency for the first pass was denoised with neighborhood information (Equation (12.6)). The corresponding reconstruction with isotropic regularization is used to estimate surface normals for the second pass. These normals provide a better estimate than the typically assumed camera-aligned direction used in classical stereo matching. The normals from the first pass further improve photometric scores and fine details (e.g. the thumb) are better preserved due to the anisotropic regularization. The last column compares textured meshes of the results in column 2 and 4. ($|V \times T| = 256^3 \cdot 3$)

regularization preserves fine details like the thumb also for a higher surface smoothness.



input and close ups | Jancosek and Pajdla [122] | PMVS [91] + Poisson [129] | isotropic model w/o normals [7] | anisotropic model with normals [8]

Figure 12.3.: Comparison of our results to other methods. Two views (top/bottom row) of the 'kick one' scene [121] (frame 1) next to an input image and close ups on details. The reconstructions by Jancosek and Pajdla [122] miss many details like the belt and the hand of the left person and parts of both heads (see close ups). Large triangles are generated at locations with low photometric information (bottom row). In contrast, the Poisson reconstruction hallucinates balloonish structures at such locations. The isotropic method in Chapter 11 yields similar results to the proposed one, but misses fine details like the belt or the hair which is difficult to recover because of noisy photometric information. The proposed normal integration yields superior results. ($|V \times T| = 256^3 \cdot 3$)

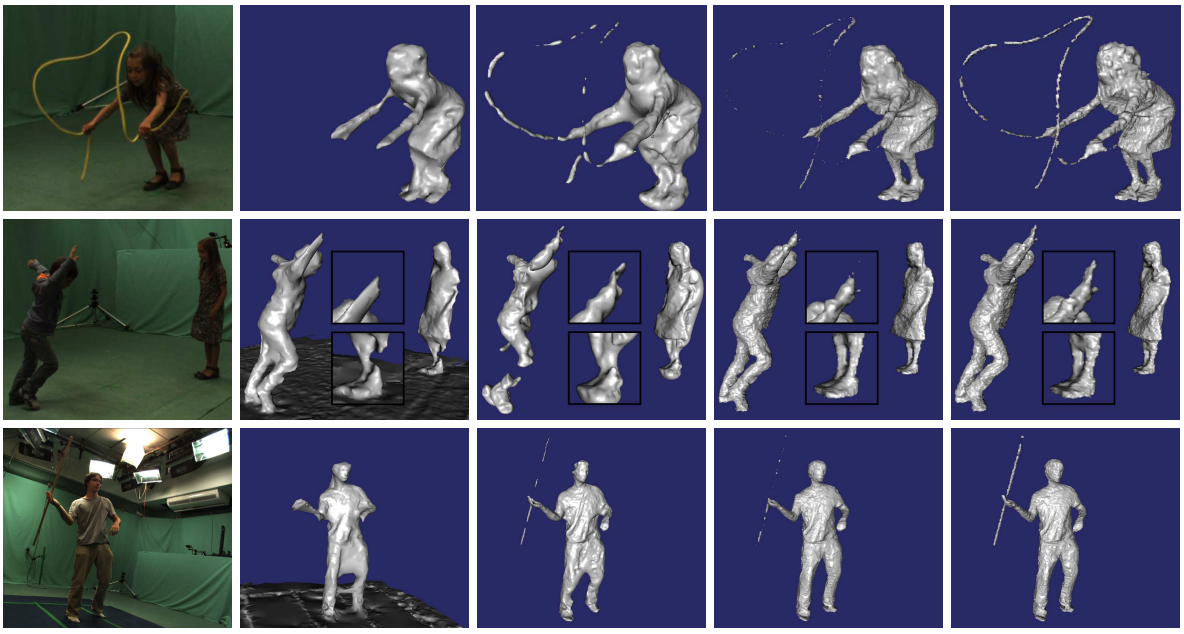
In Figure 12.3 we show reconstruction results on a martial art scene in comparison. The

method by Jancosek and Pajdla [122] tends to misconnect points which are close but not related to each other. The reconstruction of the left person shows many details on the front, because the method found many inlier points. However, the backside of the left person and most of the right person contains only few triangles which heavily degrade the visual perception of the reconstruction. Generally, this method fails to reconstruct small details and regions with low texture information like the hair or the over-bright cloth section on the shoulders. PMVS [91] performs mostly well in recovering fine details. Since PMVS is a point cloud-based method, point connectivity information is not available for the subsequent Poisson surface reconstruction [129]. This leads to misconnected points and even balloonish surface parts in regions with low photometric information. Moreover, the iterative filtering and expansion approach of [91] in combination with [129] makes the method temporally unstable in sparse camera setups.

In comparison to [122] and [91], the isotropic reconstruction method in Chapter 11/[7] performs better but cannot fully recover the belt due to bad photometric scores as well as the isotropic regularization scheme which penalizes the surface area and tends to remove thin structures (shrinking bias).

Figure 12.4 depicts results of challenging scenes with strong motion blur such as the rope jumping girl or the man with the stick. Our method does not always recover the full geometry, but generally yields better results over full video sequences (see supplementary video [8]). Mostly, fine or elongated structures are better preserved such as the fingers of the boy in the cartwheel sequence. Especially, the proposed normal-driven Gaussian smoothing in Equation (12.6) yields superior results in regions with noisy photoconsistency. In particular, the hair is consistently better reconstructed in all sequences we have evaluated. However, in areas where the photometric information is very sparse, the Gaussian smoothing can also degrade the matching score and lead to slightly worse results, e.g. the back of the person in Figure 12.3. Essentially, the reconstruction with the isotropic regularization in the first pass only serves as a smoothing of the estimated normal field. Due to the smoothing the recovered normals encode rather low-frequency details of the surface. This is in contrast to the related works mentioned in Section 12.1.1 which estimate normals to better recover the high-frequency details of the surface. However, experiments show that the estimated normals from the first pass can be estimated with moderate effort and improve the photometric matching scores in many surface regions.

Runtime. Depending on the scene, the photoconsistency and data term estimation needed about 15-30s for all 16 cameras per frame. For a volume size of $|V| = 384^3$ voxels the isotropic surface estimation in the first pass needed about 1s for $|T| = 1$ and 2-3s for $|T| = 3$. The anisotropic surface estimation in the second pass needed approximately 5s for $|T| = 1$ and 30s for $|T| = 3$. For the anisotropic regularization the optimization was 6 times faster if the normals were stored separately in a normal field instead of storing a signed distance function, but at the cost of higher memory consumption. These timings exclude loading and storing from disk and filling data structures. In comparison our method is considerably faster than PMVS [91]+Poisson [129] which needed about 20min/frame for the 'kick one' scene and 6-7min/frame for the 'cartwheel' scene. The method by Jancosek and Pajdla [122] needed about 7-10min/frame. Note that the runtime comparison is only for qualitative information, because all evaluated methods utilize CPU and GPU parallelism in a different manner and have different runtime and memory complexities. Especially the runtime of PMVS [91] is highly data-dependent because of the iterative filtering approach.



input | Jancosek and Pajdla [122] | PMVS [91] + Poisson [129] | isotropic model w/o normals [7] | anisotropic model with normals [8]

Figure 12.4.: Reconstruction results on different scenes (rope jump, boy cartwheel, stick) from [121]. Although the voxel resolution limits the quality of the rope reconstruction, normal information improves the photometric consistency and helps to better recover fine details in the matching phase and to preserve them during the surface optimization, e.g. the boys thumb or the fast moving stick or rope. Both methods [122, 91] reconstruct frames independently and show severe surface jittering. Enforcing temporal coherence visibly reduces the jittering (see supplementary video [8]) ($|V \times T| = 384^3 \cdot 3$).

12.7. Conclusion

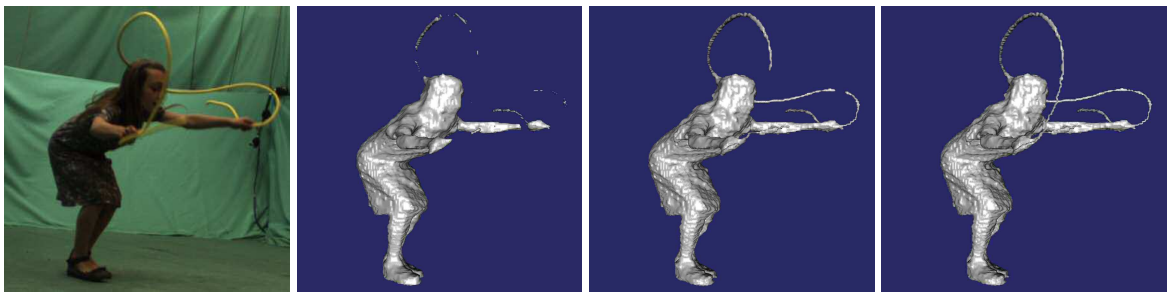
In this chapter we showed how surface normal information can be estimated and effectively used within a spatio-temporal multi-view reconstruction setup. Proper estimates of normal information firstly help to improve the accuracy of photometric measures and secondly improve reconstruction results by reducing the shrinking bias of common regularizers. Further, we demonstrated that a modification of the photoconsistency voting scheme [77] improves robustness and quality of the estimated photoconsistencies, making it more similar to methods that determine a regularized fusion of precomputed depths maps. By harnessing the power of consumer graphics cards we showed that an efficient implementation leads to low computation times despite the large amount of data being processed. Numerous experiments showed the improvements of the proposed approach over competitive reconstruction methods.

We could demonstrate that the proposed normal integration helps to recover fine elongated surface parts such as the rope or the stick in Figure 12.4. Nevertheless, the reconstructions are far from perfect, because the reconstructed rope is disconnected in many parts. In the next chapter, we will show that by enforcing the connectivity of the solution the reconstruction results can be further significantly improved.

13. Generalized Connectivity Constraints

Only through our connectedness to others can we really know and enhance the self. And only through working on the self can we begin to enhance our connectedness to others.

Harriet Lerner
(Clinical Psychologist, 1944 - present)



1 of 16 input images | No Connectivity Constraint [7] | With a Connectivity Constraint [205]+[7] | Generalized Connectivity Constraint

Figure 13.1.: Embedding connectivity constraints into multi-view reconstruction clearly helps to recover fine structures like the rope. The tree-shaped connectivity prior [205] only works for objects without holes (genus 0), resulting in disconnected parts when the rope touches the head. The proposed generalized connectivity constraint works for objects with arbitrary genus. Dataset: 'jumping rope' sequence from the INRIA 4D repository [121].

This chapter extends the spatio-temporal multi-view reconstruction framework presented in Chapter 11 and introduces connectivity preserving constraints which help to better recover fine and elongated object structures. We efficiently model connectivity constraints by precomputing a geodesic shortest path tree on the occupancy likelihood. Connectivity of the final occupancy labeling is ensured with a set of linear constraints on the labeling function. In order to generalize the connectivity constraints from objects with genus 0 to an arbitrary genus, we detect loops by analyzing the visual hull of the scene. A modification of the constraints ensures connectivity in the presence of loops. The proposed efficient implementation adds little runtime and memory overhead to the reconstruction method. Several experiments show significant improvement over state-of-the-art methods and validate the practical use of this approach in scenes with fine structured details. This work has been published in [9].

13.1. Introduction

Research in multi-view 3D reconstruction has various goals and is thus driven in many different directions. Apart from realistic physical modeling of the inverted imaging process, it is also of common interest to model learned and prior information (e.g. smoothness or shape priors), or imposing intuitive constraints on the solution, such as symmetry, connectedness or surface genus (i.e. the number of holes in the scene). In this chapter, we propose a method that is first: able to enforce connectedness of the computed solution, and second: able to preserve holes in the reconstructed scene within a multi-view reconstruction setup. We can guarantee that the solutions' surface genus is not smaller than the one of the visual hull.

Our approach is motivated by the spatio-temporal multi-view 3D reconstruction of scenes containing small object structures that we want to preserve in the reconstruction. Although fine object structures can also be preserved by incorporating exact silhouette information, such as in the work of Cremers and Kolev [62], this method is not applicable if the precomputed silhouettes are not accurate. For instance, the 3D reconstruction of a rope-jumping girl in Figure 13.1 demonstrates that a connectivity constraint on the solution helps to recover fine detailed structures like the rope. However, enforcing connectivity does not necessarily preserve holes in the reconstructed object, because such constraints only ensure that everything is connected at least once, leading to a tree shaped object structure with genus zero. The proposed generalized connectivity constraints tackle this limitation for objects of arbitrary topological genus.

13.1.1. Contributions

- We embed the concept of connectivity constraints for image segmentation into a spatio-temporal multi-view reconstruction setup.
- Since the connectivity constraints proposed in [205] only work well for scenes and objects of genus zero, we propose a generalization of the connectivity constraints to an arbitrary genus.
- We suggest an efficient implementation of the generalized connectivity constraints with a small additional memory footprint and an almost unchanged computation runtime per optimization iteration. The necessary preprocessing only adds about one minute to the three minutes computation time per frame for the presented experiments.

13.1.2. Related Work on Connectivity Constraints

The basis of this work is the spatio-temporal reconstruction method presented in Chapter 11 This method is a generalization of the 3D reconstruction by Kolev et al. [135] to the temporal domain. Both approaches use a volumetric representation of the surface within an energy minimization framework which makes it easy to impose additional constraints on the solution.

To the best of our knowledge the only previous work on connectivity in 3D reconstruction is the work of Bleyer et al. [23], in which the authors propose to use connectivity information for joint stereo matching and object segmentation. In contrast to our work, this method is rather a 2.5D than a 3D or even a 4D reconstruction method. While the authors in [23] correctly define connectivity as the existence of a connecting path, they instead propose to determine the connectivity of a pair of points by testing along a straight line that connects both points, thus only favoring convexity of objects.

In the field of image segmentation, topology preserving extensions have been proposed in different algorithmic frameworks. For the graph cut [30] algorithm, Zeng et al. [251] proposed a topology preserving refinement scheme. Chen et al. [53] propose to alternate between estimating a graph cut segmentation and modifying the respective unaries based on a level-set representation in order to fulfill predefined topological constraints. In contrast to our approach, this method does not compute minimal geodesic connections with respect to the input data and its runtime is much longer due to the iterative optimization. For the level set method a topology preserving extension was proposed by Han et al. [105]. Vicente et al. [228] use connectivity priors for a Markov random field segmentation. The authors propose an approximation scheme to enforce connectivity of the segmented object with respect to user given seed points. The drawback of all methods on connectivity mentioned so far is that they only converge to a local minimum and therefore depend on the initialization. Moreover,

apart from Bleyer et al. [23] all approaches are made for 2D domains.

Recently, three different globally optimal approaches were proposed. One is the work of Nowozin and Lampert [167], in which the constrained image segmentation problem is formulated as a linear programming relaxation. The drawback of this method is that the complexity does not scale well with the image size and therefore prevents its use for 3D or 4D reconstruction methods where the problem size easily reaches thousands or even millions of variables.

A closely related work is that of Gulshan et al. [103]. The foreground segment is restricted to the shape of a geodesic star with respect to a geodesic distance measure that depends on the image gradient. By placing several input seeds, this constraint allows several geodesic star shaped objects, their union is called a geodesic forest. However, the authors only present results on 2D image data and because the method is formulated in a graph-cut segmentation framework the boundary length regularizer is affected by the discretization.

Another globally optimal segmentation method with connectivity constraints is the work of Stühmer et al. [205]. The authors propose a geodesic tree-shaped connectivity prior for image segmentation in an efficient convex optimization framework that allows the segmentation of large scale problems as they arise for example in 3D medical imaging data. In contrast to [103], this method is formulated using a continuous segmentation framework and does not suffer from discretization artifacts with respect to the boundary length regularizer. It is perfectly suited to accurately segment objects with fine detailed tree-like structures, such as blood vessels in angiography, or the legs of insects in photographs. They first compute a single-source geodesic shortest path tree based on the image data. Then, the tree-connected segmentation is computed by imposing linear constraints on the solution, based on the precomputed shortest path tree. As such, these constraints only impose connectivity for objects without any holes or loops (genus 0).

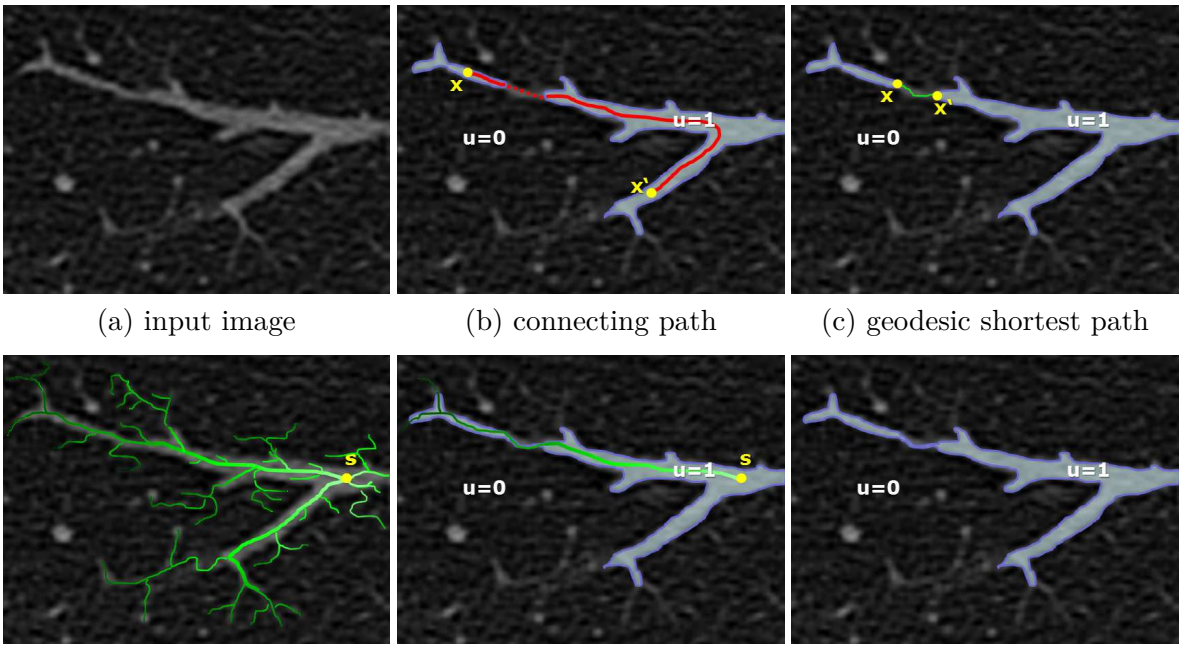
We follow this idea in the context of spatio-temporal multi-view reconstruction and generalize the connectivity constraint to objects with arbitrary genus.

13.2. Review of Connectivity Constraints for Image Segmentation

In this section, we briefly review the connectivity constraints by Stühmer et al. [205] which will be the basis for a generalized notion of connectivity in the context of spatio-temporal 3D reconstruction. In their work they introduced an efficient method for ensuring the connectedness of one region in the context of variational binary image segmentation. To sketch the main idea of [205] we consider a segmentation model which is similar to our 3D reconstruction energy. We use the same notation, but for simplicity we only consider the 2D case in this section. The segmentation problem is modeled by the binary labeling function $u : \Omega \subset \mathbb{R}^2 \to \{0, 1\}$ indicating either a foreground or a background label in each point of the image domain Ω. Hence, the labeling function u splits the image domain into two disjunct sets $\Omega_{u=1} \cup \Omega_{u=0} = \Omega$, $\Omega_{u=1} \cap \Omega_{u=0} = \emptyset$. As an example, consider the image of a blood vessel in Figure 13.2(a) in which we want to separate the blood vessel from the background. The goal is to ensure the connectivity of the foreground region $\Omega_{u=1}$ during the segmentation task. For a mathematical description of this task we make use of the following definition.

Definition 13.1 (Connectivity of a subset)**.** *Let $C_{\boldsymbol{x}}^{\boldsymbol{x}'} : [0, 1] \to \Omega$ be a continuously connected curve between two points $\boldsymbol{x}, \boldsymbol{x}' \in \Omega$ with $C_{\boldsymbol{x}}^{\boldsymbol{x}'}(0) = \boldsymbol{x}$ and $C_{\boldsymbol{x}}^{\boldsymbol{x}'}(1) = \boldsymbol{x}'$. The subset $\Omega_{u=1} \subset \Omega$ is called connected, if for any two points $\boldsymbol{x}, \boldsymbol{x}' \in \Omega_{u=1}$ there exist a path $C_{\boldsymbol{x}}^{\boldsymbol{x}'}$ between $\boldsymbol{x}$ and $\boldsymbol{x}'$ which is entirely contained in $\Omega_{u=1}$, that is $C_{\boldsymbol{x}}^{\boldsymbol{x}'} \subset \Omega_{u=1}$.*

An example of a disconnected path $C_{\boldsymbol{x}}^{\boldsymbol{x}'}$ between two points $\boldsymbol{x}, \boldsymbol{x}' \in \Omega_{u=1}$ in the foreground region is depicted in Figure 13.2(b). This definition of connectivity can directly be used to



(a) input image (b) connecting path (c) geodesic shortest path

(d) shortest path tree (e) example path on tree (f) final segmentation

Figure 13.2.: Illustration of the connectivity constraints for image segmentation. (a) Image of a blood vessel to be segmented. (b) Example path $C_{\boldsymbol{x}}^{\boldsymbol{x}'}$ between points $\boldsymbol{x}, \boldsymbol{x}'$ through disconnected foreground regions. (c) Geodesic shortest path between two separate foreground regions. (d) Illustration of the geodesic shortest path tree being grown from source node $\boldsymbol{s}$ through the entire image. (e) Example path on the tree. The constraint forces the labeling function u to grow on all paths towards the source node $\boldsymbol{s}$, thus ensuring connectivity of the foreground set $\Omega_{u=1}$. (f) Exemplary final connected segmentation. More fine, connected structures can be obtained by adjusting the smoothness parameter λ in Equation (13.2).

constrain the following variational binary image segmentation problem

$$\begin{aligned} \min_{u \in \mathcal{BV}(\Omega,\{0,1\})} \quad & \int_{\Omega} |\nabla u| \ d\boldsymbol{x} + \lambda \int_{\Omega} f u \ d\boldsymbol{x} \\ \text{s.t.} \quad & \forall \boldsymbol{x}, \boldsymbol{x}' \in \Omega_{u=1} : \exists C_{\boldsymbol{x}}^{\boldsymbol{x}'} \subset \Omega_{u=1} \ . \end{aligned} \tag{13.1}$$

Unfortunately, this constrained optimization problem is NP-hard (cf. Vicente et al. [228]) and efficient minimization is difficult. To get around this problem, the idea is to analyze which change to the labeling function u connects separated foreground regions and adds a minimum amount of cost to the energy in Equation (13.1). The answer is: It is the minimal geodesic path that connects the two regions - see Figure 13.2(c). The key idea of Stühmer et al. [205] for an efficient computation of problem (13.1) is to precompute these geodesic paths with respect to a given source point $\boldsymbol{s} \in \Omega_{u=1}$ in the foreground region. This constitutes a geodesic shortest path tree which is grown from source point $\boldsymbol{s}$ and spreads the entire image - this is sketched in Figure 13.2(d). If one now enforces labeling function u to grow towards the source node $\boldsymbol{s}$ along all geodesic paths on the tree, the connectivity of all foreground regions along any path is automatically ensured. Stühmer et al. [205] proposed the following approximation of problem (13.1) by enforcing a negative directional derivate along all paths on the precomputed geodesic shortest path tree.

$$\begin{aligned} \min_{u \in \mathcal{BV}(\Omega,\{0,1\})} \quad & \int_{\Omega} |\nabla u| \ d\boldsymbol{x} + \lambda \int_{\Omega} f u \ d\boldsymbol{x} \\ \text{s.t.} \quad & \delta_e \left(u(\boldsymbol{x}, t) \right) \leq 0, \quad e \in \mathcal{E} \end{aligned} \tag{13.2}$$

The advantage of this approach is the simplicity of the constraints - they are all linear constraints and can be efficiently handled within the optimization process. Figure 13.2(e) shows an example path which illustrated the effect: a change of the labeling function u from an interior label (≈ 1) to an exterior label (≈ 0) as shown in Figure 13.2(b) is disallowed by the constraints. This finally leads to a segmentation in which all foreground regions are connected - see Figure 13.2(f).

13.3. 3D Reconstruction with Connectivity Constraints

In this section we discuss how to integrate the connectivity constraints of Stühmer et al. [205] into our spatio-temporal multi-view method (Chapter 11). The combination of both methods allows image-based globally optimal 3D reconstruction while preserving connectivity of the object. As shown later in the experiments, this constraint also helps to reconstruct fine scale details of the scene.

For spatio-temporal multi-view reconstruction we consider exactly the same model as proposed in Chapter 11. That is, we aim to minimize the following energy.

$$E(u) = \int\limits_{V \times T} \Big(\rho |\nabla_{\boldsymbol{x}} u| + g_t |\nabla_t u| \Big) \; d\boldsymbol{x}dt + \lambda \int\limits_{V \times T} f u \; d\boldsymbol{x}dt \tag{13.3}$$

where $\lambda > 0$ controls the smoothness of reconstructed hypersurface and all other variables and functions are defined similarly as in Chapter 11.

Now, we will combine the connectivity constraints with our reconstruction energy and define all necessary variables and terms. Without loss of generality we assume that the visual hull is connected. For the case that is not connected, the same approach can be applied component-wise after identifying independent connected components of the visual hull. We define connectivity constraints independently for each time step to allow topology changes between time steps. For better readability we drop the temporal dependency in the following notation.

Graph Structure. For every time step we define a geodesic shortest path tree $\mathcal{G}_{\boldsymbol{s}}$ on the visual hull $\mathcal{VH}$ with respect to a given source node $\boldsymbol{s}$ that contains for each point $\boldsymbol{x} \in \mathcal{VH}$ inside the visual hull the shortest geodesic path $C_{\boldsymbol{s}}^{\boldsymbol{x}}$ from $\boldsymbol{s}$ to $\boldsymbol{x}$ that minimizes the cost function

$$\mathcal{D}_{\boldsymbol{s}}(\boldsymbol{x}) = \ell(C_{\boldsymbol{s}}^{\boldsymbol{x}}) = \int\limits_0^1 e^{f(C_{\boldsymbol{s}}^{\boldsymbol{x}}(r))} dr \;\; , \tag{13.4}$$

which is a positive geodesic measure that depends on the data term. Variable r parametrizes the path from $\boldsymbol{s}$ to $\boldsymbol{x}$. $\mathcal{D}_{\boldsymbol{s}}(\boldsymbol{x})$ is a shorthand for the distance map of the shortest geodesic path from the source node $\boldsymbol{s}$ to any point $\boldsymbol{x} \in \mathcal{VH}$. The edges of the shortest paths form the edge set $\mathcal{E}$ of the shortest path tree $\mathcal{G}_{\boldsymbol{s}}$.

Source Node Computation. It is desirable to center the source node for the geodesic shortest path computation within the data term. To this end, we compute the source node $\boldsymbol{s}(t)$ as the point which minimizes a spatio-temporal convolution of the data term f with a sufficiently large Gaussian kernel $\mathcal{G}$.

$$\boldsymbol{s}(t) = \arg\min_{\boldsymbol{x}} \int\limits_{t-1}^{t+1} (f * \mathcal{G})(\boldsymbol{x}, \tau) \, d\tau \tag{13.5}$$

The minimization reflects the fact that negative data term values $f < 0$ indicate a favor for an interior label and thus ensures a position that has high probability of being interior. The position of the source node has not much influence on the result, but this choice favors a smoothly temporal change of its position within the data term while maximizing the distance to the surface. An example rendering of a shortest path from a leaf node to the source is shown in Figure 13.5a.

Constrained Optimization. The connectivity constraint from [205] is included into the reconstruction process as a monotonicity constraint of the labeling function u with respect to the edges $\mathcal{E}$ in $\mathcal{G}_{\boldsymbol{s}}$. This monotonicity can be ensured by including inequality constraints on the directional derivative $\delta_e\left(u(\boldsymbol{x},t)\right)$ of u along every edge $e \in \mathcal{E}$. Thus, computing a spatio-temporal 3D reconstruction with connectivity constraints can be achieved by computing a minimizer of the constrained optimization problem

$$\begin{aligned} \min_{u \in \mathcal{BV}(V \times T, \{0,1\})} \quad & E(u) \\ \text{s.t.} \quad & \delta_e\left(u(\boldsymbol{x},t)\right) \leq 0, \quad e \in \mathcal{E} \end{aligned} \tag{13.6}$$

with one constraint for each edge e in the edge set $\mathcal{E}$ of the shortest path tree $\mathcal{G}_{\boldsymbol{s}}$. $\mathcal{BV}(\cdot)$ denotes the function space of bounded variations [13] - see Section 2.3.

13.4. Generalized Connectivity Constraints for Objects of Arbitrary Genus

The key idea to generalize the connectivity constraint to objects with arbitrary genus is a modification of the constraints that are defined on the geodesic shortest path tree. The key ingredient to this modification is to detect loops in the object and to identify parts of these loops with a 'thin' geometry, called handles. This is described in the following.

13.4.1. Handle and Tunnel Loops

In [73], Dey et al. study arbitrary surfaces represented by a simplicial complex, that is, a hierarchy of p-simplicies with different dimensions p (e.g. $p = 0, \ldots, 2$ corresponding to points, edges, and faces). The surface $\mathbb{M}$ separates the simplicial complex into an interior part $\mathbb{I}$ and an exterior part $\mathbb{E}$, both including the surface, i.e. $\mathbb{I} \cap \mathbb{E} = \mathbb{M}$. Since we want to analyze the topology of the visual hull, these sets will be shorthands for $\mathbb{M} = \partial\mathcal{VH}$, $\mathbb{I} = \mathcal{VH}$ and $\mathbb{E} = (V \setminus \mathcal{VH}) \cup \partial\mathcal{VH}$.

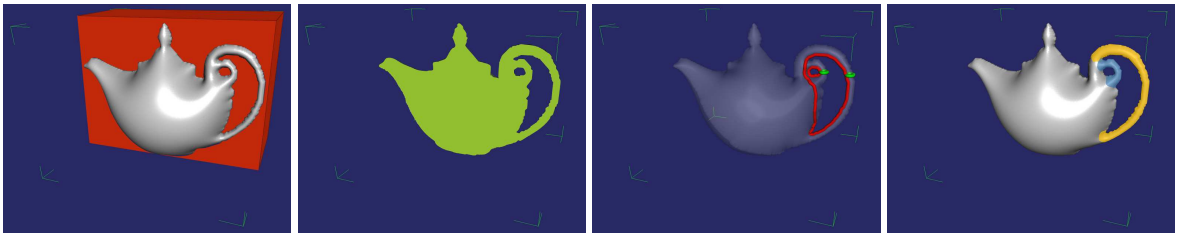
The authors in [73] define and study cycles of edges ('loops') on the surface which build equivalence classes with respect to contraction or translation of the cycle - like a rubber band which can be moved along the surface, but not above holes in the surface. In this chapter we call this equivalence relation $\sim_{\mathbb{M}}$ 'contractible' on the set $\mathbb{M}$, for example, we denote the relation that a loop $l_1 \subset \mathbb{M}$ is contractible to a loop $l_2 \subset \mathbb{M}$ on the set $\mathbb{M}$ as $l_1 \sim_{\mathbb{M}} l_2$. For simplicity we try to define terms and notation on a more intuitive level which should be sufficient to follow the rest of the chapter. For mathematically precise definitions based on persistent homology we refer to [73]. Following their work, we now consider loops on the surface with the following properties.

Definition 13.2 (Handle and tunnel loops)**.** *A* ***handle loop*** $h \subset \mathbb{M}$ *is a cycle of edges on the surface that is contractible in the interior ($h \sim_{\mathbb{I}} 0$) and not contractible on the surface*

*($h \nsim_{\mathbb{M}} 0$). A **tunnel loop** $t \subset \mathbb{M}$ is a cycle of edges on the surface that is contractible in the exterior ($h \sim_{\mathbb{E}} 0$) and not contractible on the surface ($h \nsim_{\mathbb{M}} 0$).*

With respect to the above mentioned equivalence relation, a closed surface of genus g has exactly g classes of handle loops and g classes of tunnel loops induced by the surface embedding. We consider one representative loop with approximate minimal geometric length per class and denote them as the set of handle loops $\{h_i\}_{i=1}^{g}$ and the set of tunnel loops $\{t_i\}_{i=1}^{g}$. Hence, for each surface hole i we have a corresponding pair (h_i, t_i) of representative handle and tunnel loops.

Examples of handle and tunnel loops are shown in Figures 13.3 and 13.4 and Figure 13.5c. Dey et al. [73] also propose an algorithm which computes handle and tunnel loops with approximate minimal length that is perfectly suited to process volumetric data. However, this algorithm is considerably slower than a recently published algorithm by Dey et al. [72] which only works for meshes. To this end, we extract an iso-surface mesh of the visual hull to efficiently compute handle and tunnel loops. The speed advantage of the method in [72] stems from the fact that it does not need a 3D tessellation of the scene. In [72], the concept of Reeb graphs is used to estimate an initial set of handle and tunnel loops and their geometric length is shortened in a subsequent refinement step.



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

Figure 13.3.: Various sets defined in this section visualized on a teapot model of genus 2. (a) Exterior $\mathbb{E}$ (red), (b) Interior $\mathbb{I}$ (green), (c) Handle and tunnel loops $\{h_1, h_2\}, \{t_1, t_2\}$ (green+red), (d) Handle segments H_1, H_2 (yellow+blue).

Handle Segmentation. We aim to segment the 'thin' geometric parts around the holes of the surface, called handles. These handle segments will help to make the connectivity constraints adaptive to the data term. For this purpose we introduce the following definitions.

Definition 13.3 (Handle Segment Surface)**.** *We define the handle segment surface as the connected subset of all points $\boldsymbol{x} \in \mathbb{M}$ for which a handle loop $h_{\boldsymbol{x}}$ exists which is contractible to h_i subject to the additional constraint that the ratio of $\ell(h_{\boldsymbol{x}})$ and $\ell(h_i)$ does not exceed a user given threshold σ:*

$$\mathbb{M}_{H_i} = \left\{ \boldsymbol{x} \in \mathbb{M} \;\middle|\; \exists h_{\boldsymbol{x}} \subset \mathbb{M} : h_{\boldsymbol{x}} \sim_{\mathbb{I}}^{\sigma} h_i \right\} \tag{13.7}$$

where $h_{\boldsymbol{x}} \subseteq \mathbb{M}$ denotes a handle loop through the surface point $\boldsymbol{x}$ and $h_{\boldsymbol{x}} \sim_{\mathbb{I}}^{\sigma} h_i$ means that handle loop $h_{\boldsymbol{x}}$ is contractible to h_i subject to the constraint $\ell(h_{\boldsymbol{x}}) < \sigma \ell(h_i)$.

Definition 13.4 (Handle Segment)**.** *Given the handle segment surface $\mathbb{M}_{H_i}$ from the previous definition, we define the corresponding volumetric handle segment $H_i \subseteq \mathbb{I}$ as the set of all points in the visual hull for which the closest point on the visual hull boundary is on the handle segment surface $\mathbb{M}_{H_i}$.*

$$H_i = \left\{ \boldsymbol{x} \in \mathbb{I} \;\middle|\; \operatorname*{arg\,min}_{\boldsymbol{y} \in \mathbb{M}} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{M}_{H_i} \right\} \tag{13.8}$$

where $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y})$ *denotes the Euclidean distance between point* $\boldsymbol{x} \in \mathbb{I}$ *in the interior and point* $\boldsymbol{y} \in \mathbb{M}$ *on the surface.*

In practice, we compute H_i by a breadth first search algorithm on the visual hull. Starting from the handle loop h_i a wavefront is propagated in both directions. Independently for each wavefront, we stop the search if the ratio between the current length of the wavefront and the initial position exceeds the threshold σ.

13.4.2. Loop Connectivity Constraints

With the handle and tunnel loops of the visual hull we are now able to generalize the connectivity constraint in the presence of loops. By enforcing interior labels along each tunnel loop t_i we can assure that loops in the visual hull are preserved in the final segmentation. However, in order add a minimum amount of costs to the energy in Equation (13.6) when enforcing loop connectivity, we need to find corresponding loops that respect the costs of the data term. We approximate these geodesics shortest loops by computing corresponding loops $t_i^{\mathcal{G}_{\boldsymbol{s}}} \subset \mathbb{I}$ on the precomputed geodesic shortest path tree $\mathcal{G}_{\boldsymbol{s}}$ which are contractible to the original tunnel loop on the surface, i.e. $t_i^{\mathcal{G}_{\boldsymbol{s}}} \sim_{\mathbb{I}} t_i$. The computation of $t_i^{\mathcal{G}_{\boldsymbol{s}}}$ is discussed later in this section. For each tunnel loop t_i of the visual hull we define a *loop preserving* constraint as

$$\forall i \in [1, \ldots, g] : \quad \left\{ \forall \boldsymbol{x} \in t_i^{\mathcal{G}_{\boldsymbol{s}}} : u(\boldsymbol{x}) = 1 \right\}. \tag{C0}$$

Proposition 13.5. *The constraint* (C0) *preserves the handle and tunnel loops and thus all holes of the visual hull in the reconstructed object. The topological genus of the reconstructed object is larger or equal to the one of the visual hull.*

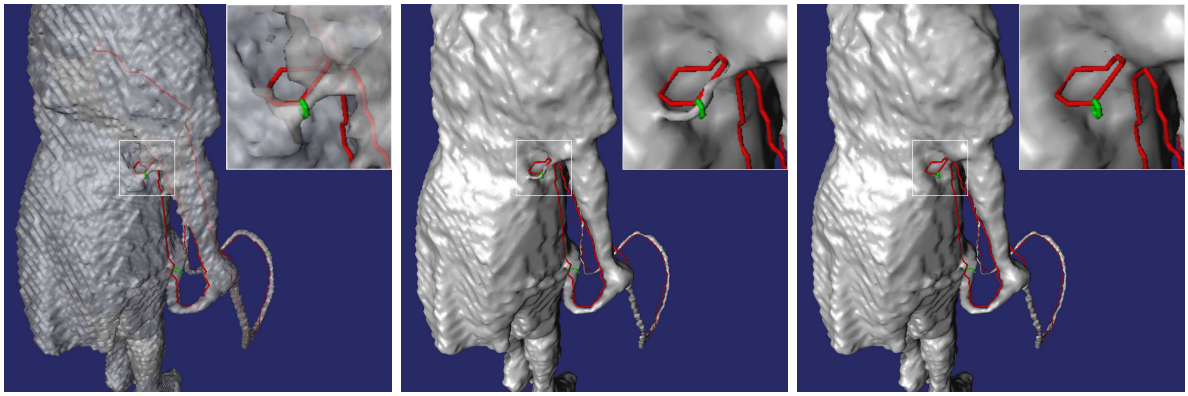
Proof. Let us assume that the proposition does not hold. To let the genus of the reconstructed object decrease, either (i) at least one hole of the visual hull needs to be filled or (ii) at least one tunnel loop has to be disconnected in the reconstructed object. Because the domain of the reconstructed object is restricted to the visual hull, (i) cannot be fulfilled. By construction, (ii) is fulfilled if (C0) is fulfilled. Therefore the genus of the reconstructed object has to be larger or equal to the genus of the visual hull. □

Note that, depending on the data term f the reconstructed object is allowed to have more holes than the visual hull. In some cases, it is not desirable to exactly preserve all holes and corresponding handles of the visual hull. A possible scenario is depicted in Figure 13.4 where aliasing artifacts of the visual hull lead to spurious handle loops which should not be preserved in the final reconstruction. Therefore we propose to relax the loop preserving constraint (C0) such that either the connectivity of a handle is preserved in the final reconstruction or, in case the photometric support via f is not strong enough, the handle segment H_i is suppressed completely. We define the *generalized connectivity* constraint as

$$\forall i \in [1, \ldots, g] : \quad \left\{ \forall \boldsymbol{x} \in t_i^{\mathcal{G}_{\boldsymbol{s}}} \cap H_i : \frac{d}{ds} u(\boldsymbol{x}) = 0 \right\} \tag{C1}$$

where $\frac{d}{ds}$ is the directional derivative along the loop $t_i^{\mathcal{G}_{\boldsymbol{s}}}$.

Finding the optimal connected loop $t_i^{\mathcal{G}_{\boldsymbol{s}}}$. For objects of genus 0, the use of the shortest path tree in the connectivity constraint is motivated by the optimal connecting path, that adds the minimum cost to the final segmentation result. In case of objects with higher genus, we wish to preserve the connectivity with respect to loops in the final segmentation. Therefore



(a) Visual Hull

(b) Result with strict loop connectivity (C0)

(c) Result with generalized connectivity (C1)

Figure 13.4.: Comparison of the two connectivity constraints. (a) In some cases artifacts of the visual hull can lead to spurious handle loops which should not be preserved in the final reconstruction. (b) The constraint C0 strictly preserves all loops in the solution. (c) Relaxing the topology preserving constraint to our generalized connectivity constraint allows to suppress handles where the photoconsistency is not strong enough. The rope, where the support of the photoconsistency is sufficient, is still completely preserved. Handle and tunnel loops are depicted in green and red, respectively.

a loop through each handle needs to be found, which is optimal in the same way, i.e. that it also adds the minimum cost to the final segmentation. Using the already computed shortest path tree $\mathcal{G}_{\boldsymbol{s}}$, we can find the shortest loop $t_i^{\mathcal{G}_{\boldsymbol{s}}}$ with respect to $\mathcal{G}_{\boldsymbol{s}}$ for each handle i by the following steps: With a depth first search on $\mathcal{G}_{\boldsymbol{s}}$, starting from the boundary of a handle segment H_i, we compute the partitions $H_i^1 \cup H_i^2 = H_i, H_i^1 \cap H_i^2 = \emptyset$ which are disconnected on the shortest path tree $\mathcal{G}_{\boldsymbol{s}}$. These partitions are shown in Figure 13.5d. If one of these partitions is empty, i.e. all points in the handle segment H_i are connected on $\mathcal{G}_{\boldsymbol{s}}$, then no further constraints need to be added in order to preserve handle segment H_i. Otherwise, we compute an optimal pair of points

$$(\boldsymbol{p}, \boldsymbol{q}) = \underset{(\boldsymbol{x} \in H_i^1, \boldsymbol{y} \in H_i^2, \boldsymbol{y} \in \mathcal{N}(\boldsymbol{x}))}{\arg\min} \mathcal{D}_{\boldsymbol{s}}(\boldsymbol{x}) + \mathcal{D}_{\boldsymbol{s}}(\boldsymbol{y}) \tag{13.9}$$

which are leaf-nodes in $\mathcal{G}_{\boldsymbol{s}}$. The set $\mathcal{N}(\boldsymbol{x})$ denotes the local spatial neighborhood of a point $\boldsymbol{x} \in V$. The optimal path through the handle is computed by tracing the path backwards along the predecessors of both nodes $\boldsymbol{p}, \boldsymbol{q}$ in $\mathcal{G}_{\boldsymbol{s}}$, resulting in the path with minimum costs through the handle (Figure 13.5e).

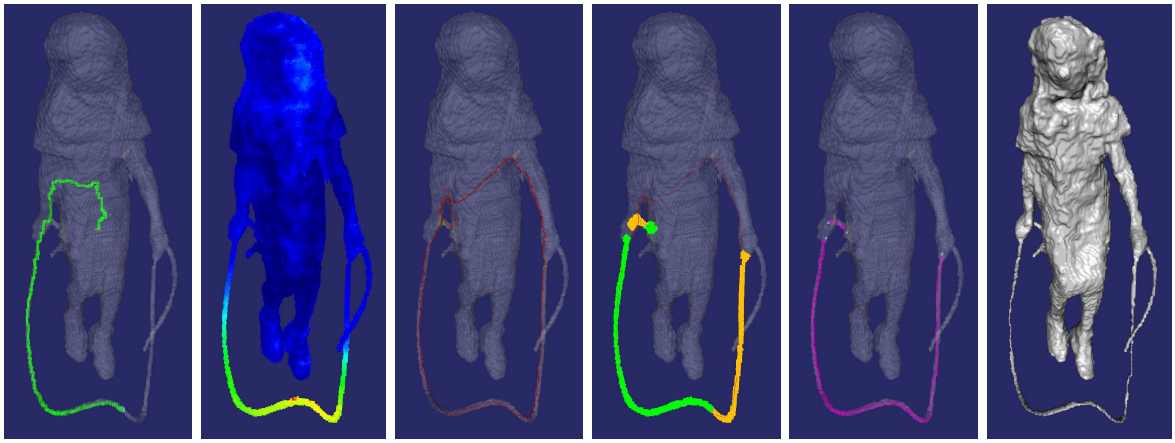
While the tree connectivity constraint resulted in an inequality constraint on the derivative of the label function, the loop connectivity is preserved by adding the equality constraints

$$\delta_e \left(u(\boldsymbol{x}, t) \right) = 0, \quad e \in \mathcal{E}_{=}. \tag{13.10}$$

to the optimization problem in Equation (13.6), where $\mathcal{E}_{=}$ is the set of edges of the optimal path through the handle.

13.5. Numerical Optimization

To minimize energy (13.6) using convex optimization we first relax the discrete image function to the continuous interval $[0, 1]$. The constraints defined on the derivative of the image function remain the same as in the discrete setting.



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

Figure 13.5.: Visualization of various properties that we compute based on the shape of the visual hull (genus 2 in this case) and the data term. (a) Example shortest path from a leaf node to the source node $\boldsymbol{s}$ (red); (b) color-coded geodesic distance map $\mathcal{D}_{\boldsymbol{s}}$ with respect to the source node $\boldsymbol{s}$; (c) handle (green) and tunnel (red) loops; (d) handle segmentations $H_i = H_i^1 \cup H_i^2$ (green+orange), the coloring shows disconnected parts within the handle with respect to the geodesic path tree $\mathcal{G}_{\boldsymbol{s}}$. (e) shortest path through the handle for which the equality constraints (C1) are imposed; (f) final reconstruction result.

Because the total variation norm is non-differentiable, we introduce a dual variable $\boldsymbol{p} : V \times T \to \mathbb{R}^4$ and reformulate the optimization problem Equation (13.6) as the equivalent saddle-point problem

$$\begin{aligned} \min_{u} \max_{\|\boldsymbol{p}\| \leq 1} & \int_{V \times T} \langle u, -\operatorname{div}(\boldsymbol{p}) \rangle \; d\boldsymbol{x}dt + \lambda \int_{V \times T} f u \; d\boldsymbol{x}dt \; . \\ \text{s.t.} \quad & \delta_e \left(u(\boldsymbol{x}, t) \right) \leq 0, \quad e \in \mathcal{E} \\ & \delta_e \left(u(\boldsymbol{x}, t) \right) = 0, \quad e \in \mathcal{E}_= \end{aligned} \tag{13.11}$$

The constraints on u over the edge sets $\mathcal{E}$ and $\mathcal{E}_=$ are included in the optimization using Lagrangian multipliers β and γ. The Lagrangian associated to problem (13.11) becomes

$$\begin{aligned} \min_{u} \max_{\substack{\|\boldsymbol{p}\| \leq 1, \\ \beta \geq 0, \\ \gamma}} & \int_{V \times T} \langle u, -\operatorname{div}(\boldsymbol{p}) \rangle \; d\boldsymbol{x}dt + \lambda \int_{V \times T} f u \; d\boldsymbol{x}dt \\ & + \int_{T} \left\{ \sum_{e \in \mathcal{E}} \beta_e \, \delta_e \left(u \right) + \sum_{e \in \mathcal{E}_=} \gamma_e \, \delta_e \left(u \right) \right\} dt \; . \end{aligned} \tag{13.12}$$

This saddle point problem is optimized using the preconditioned primal-dual algorithm by Pock and Chambolle [175]. The algorithm results in an iterative update scheme with a

gradient ascent in the dual and a gradient descent in the primal variable

$$\begin{aligned}
\boldsymbol{p}^{k+1} &= \Pi_C \left[\boldsymbol{p}^k + \sigma \nabla \bar{u}^k\right] \\
\beta_e^{k+1} &= \Pi_{\geq 0} \left[\beta_e^k + \mu\, \delta_e \left(\bar{u}^k\right)\right] \\
\gamma_e^{k+1} &= \gamma_e^k + \nu\, \delta_e \left(\bar{u}^k\right) \\
u^{k+1} &= \Pi_{[0,1]} \left[u^k + \tau\Big(\operatorname{div} \boldsymbol{p}^{k+1} + \operatorname{div} \beta^{k+1} + \operatorname{div} \gamma^{k+1} - \lambda f\Big)\right] \\
\bar{u}^{k+1} &= 2u^{k+1} - u^k
\end{aligned} \tag{13.13}$$

where $\Pi_{[0,1]}$ is the projection of u onto the unit interval $[0, 1]$ and $\Pi_{\geq 0}$ onto positive values. The projection onto the set $C = \{q = (q_{\boldsymbol{x}}, q_t)^T : V{\times}T \to \mathbb{R}^4 | \, \|q_{\boldsymbol{x}}\| \leq 1, |q_t| \leq 1\}$ is a projection on a 4D hyperball and can be done as follows:

$$\Pi_C(q) = \left(\frac{q_{\boldsymbol{x}}}{\max(1, \frac{\|q_{\boldsymbol{x}}\|}{\rho})}, \max\left(-g_t, \min(g_t, q_t)\right)\right)^T \tag{13.14}$$

The step sizes τ, σ, μ and ν are chosen as suggested in [175]. Because our energy model is convex and the linear constraints preserve convexity of the optimization problem, the update scheme (13.13) converges to a global minimum of the relaxed energy (13.6). An optimal binary labeling can be found by thresholding the relaxed solution [175].

Implementation. The proposed iterative scheme for minimal surface reconstruction with connectivity constraints (13.13) allows a high degree of parallelization and is implemented using the CUDA programming framework. The connectivity graph precomputation is more difficult to parallelize and therefore is implemented on the CPU.

13.6. Experiments

We evaluated our method on several spatio-temporal multi-view data sets provided by the INRIA 4D repository [121]. All scenes were synchronously recorded by 16 cameras in a green room environment.

In the experiments we mainly focus on comparing reconstruction results with and without connectivity constraints. Since no other 4D reconstruction methods are publicly available, we compare our results with the ones of the state-of-the-art 3D reconstruction methods by Jancosek and Pajdla [122] and the combination of Furukawa et al. (PMVS) [91] and Poisson surface reconstruction [129].

Approximate silhouette information was used for all methods except of the method by Jancosek and Pajdla [122] for which it cannot be used. We used the 6-neighborhood for the computation of the geodesic shortest path tree $\mathcal{G}_{\boldsymbol{s}}$. In this setting, the generalization to arbitrary genus by using equality constraints does not increase the number of dual variables (Lagrange multipliers), because some inequality constraints are exchanged by equality constraints.

Runtime and Memory Resource Evaluation. The memory footprint of the suggested implementation increases only by $|V{\times}T|$ bytes in comparison to the original approach. The numerical optimization runtime per iteration remains almost unchanged, but depending on the scene structure more iterations are needed for sufficient convergence. All experiments

were run on a Linux-based Intel Xeon E5520 PC with 24GB RAM and NVidia GTX Titan graphics card. For the genus 0 connectivity [205] the precomputation time per frame was about 20 seconds for computing the tree of the tree-shaped connectivity constraints. For the generalized connectivity constraints the precomputation time was about 1 minute for handle and tunnel loop detection, handle segmentation and computation of the tree. The optimization needs about 3 minutes per frame resulting in a total runtime of about 4 minutes per frame when using the generalized connectivity constraints.

13.7. Conclusion

In this chapter we introduced tree-shaped connectivity constraints into spatio-temporal multi-view 3D reconstruction. By detecting loops in the object we are able to generalize the connectivity constraint to objects with non-tree structure of arbitrary genus. In several experiments, we demonstrated that the proposed connectivity constraints significantly improve the reconstruction quality in the presence of fine elongated structures.

To the best of our knowledge, apart from the work in [23], which uses a strong simplification of a connectivity prior and essentially is a 2.5D method, this is the first work which imposes connectivity constraints in a multi-view 3D reconstruction setup and provides an efficient way to enforce them.

The connectivity constraint is especially useful in 4D multi-view settings, for which exact silhouettes are usually not available and exact silhouette constraints are not applicable. Assuring temporal coherence of the connectivity constraints would need explicit modeling of the occupancy flow and remains for future work.

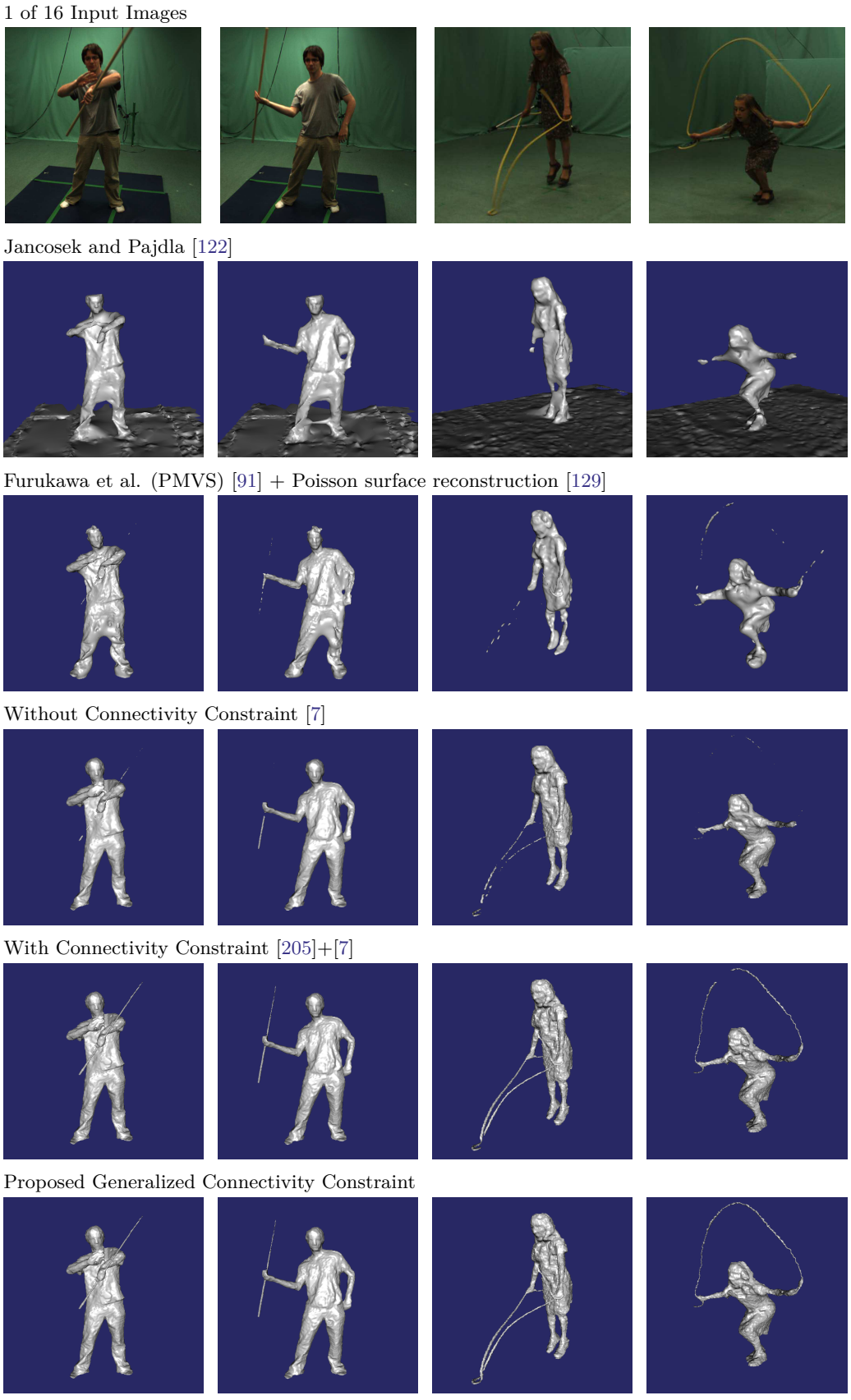


Figure 13.6.: Comparison of different reconstruction methods: Existing state-of-the art approaches [122, 91, 129] fail to recover thin structures like the stick and the rope. The connectivity constraint allows to preserve the stick, but for the rope-jump scene with higher genus, it does not completely preserve the connection of the rope. Our proposed generalized connectivity constraint allows to correctly reconstruct both scenes (volume resolution $|V| = 384^3$).

Part IV.

Conclusions and Outlook

14. Concluding Remarks

Anyone can initiate. The real skill is knowing how to "finitiate".

Devarajan (Dave) Thirumalai
(Professor of Biophysics, University of Maryland, 1955 - present)

This thesis investigated two extreme cases of 3D reconstruction: the reconstruction from a single image and the reconstruction over time from multiple-view image sequences. Although both problems exhibit very different challenges, we demonstrated that the investigated 3D reconstruction framework with a minimal surface prior is suitable to work with all cases. For both cases we presented a series of novel methods which tackle the problem-specific difficulties with various priors.

Single-View Reconstruction. In Part II, we introduced three novel approaches to user-guided single-view reconstruction. All three methods are tailored to reconstruct curved 3D objects from an input silhouette with arbitrary topology which was not possible with previous methods. All methods require significantly less user input to obtain plausible reconstructions than related methods.

In Chapter 5, we discussed the single-view reconstruction problem and gave an overview of related work. Further, we classified and compared the related work with respect to a number of method properties such as their application domain, kind of surface representation, important assumptions, their user input, as well as image cues and priors used by the method.

In Chapter 6, we introduced our single-view reconstruction framework with a minimal surface prior in combination with a novel silhouette-based shape prior for solving the surface inflation problem more elegantly and effectively than related approaches.

In Chapter 7, we showed that the shape prior can be replaced with a volume prior which further reduced the amount of user input and made it more intuitive to use. Moreover, the volume prior avoids surface discontinuities that are apparent in all shape prior-based reconstructions.

In Chapter 8, we identified that the model with the volume prior can be solved much more efficiently and accurately by giving up some flexibility and reverting to a simpler parametric surface representation.

In Chapter 9, we demonstrated that our methods compare well to other state-of-the-art methods. We further showed that our methods need significantly less user input and modeling time while obtaining comparable reconstruction results. This is mainly because our proposed inflation priors keep the amount of the user input minimal and intuitive. Further, we compared our methods to the most related ones ([117, 253, 183]) with respect to their advantages and disadvantages.

Spatio-Temporal Multi-View Reconstruction. In Part III, we proposed a novel approach to spatio-temporal multi-view reconstruction by generalizing the 3D reconstruction method by Kolev et al. [134] to the temporal domain.

In Chapter 11, we demonstrated that this generalization is non-trivial. We proposed a novel

data-term to better deal with the sparsity of typical multi-view video setups and to lower the complexity of its computation in order to make the approach usable to process longer sequences. The approach yields competitive and temporally smoother reconstruction results with shorter computation times than comparable approaches which do not even enforce temporal coherence.

In Chapter 12, we showed that our simple but efficient way for approximating surface normals helps to improve the reconstruction quality significantly, as this information can be used effectively in several places of our reconstruction approach: for improving photometric matching scores, for computing anisotropically smoothed depth-hypotheses and for regularizing the surface in an anisotropic manner. The reconstruction results showed clear improvements over the isotropic approach.

In Chapter 13, we proposed the first 3D/4D reconstruction framework which integrates connectivity constraints that are efficiently computable. Further, we were able to generalize the connectivity constraints into topological constraints which are particularly useful for spatio-temporal 3D reconstruction. Due to the constraints our method clearly outperformed state-of-the-art methods.

General 3D Reconstruction. In this thesis we further demonstrated that all cases of 3D reconstruction can be modeled within an almost unified variational 3D reconstruction approach, which is flexible, elegant and highly extendible for incorporating a variety of prior information. So far, the special cases of single-view reconstruction and spatio-temporal multi-view reconstruction have usually been studied separately in the literature, and most of these methods do not share many similarities and their extension to other reconstruction scenarios is usually not straightforward. In this sense, this work can also be seen a first step to create a general model that is able to deal with any number of input images. An important feature of such a model will be the possibility to integrate and combine a number of priors that are valid for any input scenario and most importantly help to tackle the ill-posedness of the reconstruction task. In this thesis, we extended the 3D reconstruction model by making use of a minimal surface prior, as well as symmetry, shape, volume, and connectivity priors.

15. Limitations and Future work

The scientist, by the very nature of his commitment, creates more and more questions, never fewer. Indeed the measure of our intellectual maturity, one philosopher suggests, is our capacity to feel less and less satisfied with our answers to better problems.

Gordon W. Allport
(American Psychologist, 1897 - 1967)

15.1. Single-View Reconstruction

Since the proposed 3D reconstruction framework is very universal and extendible, a rather general direction for future work is trying to include other priors, constraints or model extensions. First advances have already been successfully made by Töppe et al. [210] and Vicente and Agapito [227]. When looking at the results, the most astonishing fact about the proposed single-view approach is certainly that only silhouette information is used. Of course, the proposed depth-inference heuristics impose strong limitations on the class of objects that can be reconstructed. Obviously, more image information should be used in order to obtain better and accurate rather than pleasing reconstruction results.

One possible way of using more image information is by learning approaches which learn the relation between the appearance of image regions and for example depth values or surface normals and are then able to estimate these properties from a single input image. Recent advances on that topic by Ladický et al. [143, 144] yield promising results.

Another interesting direction is to formulate a weaker and a more general form of the symmetry constraint, because the required side-view is a rather strong assumption that limits the applicability of the approach considerably. Interestingly, there exists a single-view reconstruction method by Köser et al. [137] which requires nearly frontal views of plane symmetric objects or scenes. Since their approach together with our method cover the extreme cases of viewing angles on plane symmetric objects, it would be interesting to know whether a combination of these methods will help to handle arbitrarily oblique viewing angles.

15.2. Spatio-Temporal Multi-View Reconstruction

As we have already demonstrated in this thesis the proposed spatio-temporal reconstruction approach is easily extendible and there are many possibilities for improvements and several limitations which deserve further investigation.

Photoconsistency measure. The limitations of the Lambertian reflection model are well known and still it is widely used because of its simplicity. Thus, it belongs to the "usual suspects" for possible improvements. Nevertheless, more realistic light models easily make the corresponding optimization problem infeasible.

Another important aspect that has been widely ignored in most photometric stereo approaches is the influence of image scale on the matching process. The inherent assumption in

most 3D reconstruction methods (apart from scale-invariant feature based methods) is that all parts in the input images are perfectly in focus or at least have a similar out-of-focus blur when being compared in the matching process. Although Hornung and Kobbelt [113] already considered this issue in 2006, especially recent work on multi-view stereo by Bradley and Beeler [32] and work on optical flow estimation by Sevilla-Lara et al. [197] demonstrated a significant accuracy increase. In contrast to Bradley and Beeler [32] who select the best-matching level in the scale-space, Sevilla-Lara et al. [197] create a feature vector containing all scales and use that for matching. Both approaches are not difficult to integrate into the 3D reconstruction approach considered in this thesis.

Surface priors. As motivated in the beginning of this thesis, priors help to deal with the ill-posedness of the problem and several priors have also been identified in human vision. The challenge is to formulate them in a tractable and feasible manner. Obviously, it would be useful in some cases to also use (more general) symmetry priors or volume priors in the multi-view reconstruction case. In fact, we have implemented volume priors also into our spatio-temporal framework, the same way as for the single-view case. However, their effect was not as expected and not useful, because the compactness argument (Section 7.3) of the volume-constrained reconstruction approach relies on a proper data fidelity term and most importantly on the boundary conditions which are typically different in both setups. In the general case, the compactness of the relaxed solution cannot be guaranteed, but might be enforced by adding non-convex penalizers to the energy that repel non-binary solutions.

Regularization. The *spatial regularization* via total variation has many desirable advantages, but a major drawback is the so-called *shrinking bias* towards smaller solutions due to the surface area penalization. As a further result of the area penalization, solutions tend to be compact and fine structures and details are suppressed. As shown in Chapter 12 this effect can be reduced by using anisotropic metrics, but the general problem still persists. Curvature-based regularization (e.g. [194, 166]) seems to offer a solution to this problem, but since even weak approximations are computationally much more expensive, further research on this topic is necessary. An interesting approach to go beyond total variation, is the idea to learn the regularizer from example data. For instance, Häne et al. [106] categorize different scene parts (such as trees, houses, and streets) into classes and learn separate regularizers for each class. In a combined class segmentation and 3D reconstruction process the regularization resembles the class-specific surface properties and thus improves the reconstruction accuracy. Apart from generalizing this approach into a 4D setup, a similar approach might be useful for temporal regularization, if, for example, different repetitive motion patterns are present in the scene.

The *temporal regularization* model as proposed in this thesis is very simple and only affects static or nearly static scene parts. The proposed temporal weighting avoids artifacts that would occur in scene parts with faster motion. Instead of penalizing local scene changes it would be more meaningful to jointly estimate the motion of all scene parts and then, in turn, regularize the motion field to penalize non-smooth local deformations.

Scalability. If volumetric reconstruction approaches are discretized on a regular grid, as presented in this thesis, considerable amounts of memory are needed even for small scenes with practicable volume resolutions. In a straightforward manner, this approach does not scale well to large-scale scenarios. A solution to this problem are data adaptive domain discretizations. For example voxel octrees have been shown to scale well for a variety of approaches and have been used extensively in the literature, for depth map fusion [39, 88, 250, 202, 203], or point cloud-based surface reconstruction [129, 130], and have also been extended to a spatio-

temporal setting [215]. Other promising approaches for data-adaptive data storage include voxel hashing [165] or time adaptive storage of static scene parts [239].

Optimality. Although the estimation of the weighted minimal surface in our approach is globally optimal, the pre-computation of the data fidelity term is not. In particular, the applied voting scheme determines decisions about depth values that are potentially wrong. In this sense, our approach still has similarities to methods that fuse pre-computed depth maps as the solution of two sequential subproblems. However, in contrast to these methods, our approach also transfers matching qualities along with the depth maps in the form of probability distributions. A much better approach would be to couple these two dependent problems and solve them jointly in a single optimization approach.

Camera calibration and texture. The ultimate goal is to have a robust reconstruction approach, which estimates everything at the same time: surface geometry and motion, color or texture information and camera calibration. In the current setup, the cameras are assumed to be pre-calibrated and surface textures are computed in a post-processing step. Along the lines of works that already address joint calibration and reconstruction, such as [217, 15], or the combination of texture estimation and reconstruction as in [64, 39, 215], or both [71], their globally optimal integration in the present framework remains a challenge.

Notation

$\langle a, b \rangle$	inner product of a and b
$\mathbf{1}_A$	indicator function for set A: $\mathbf{1}_A(x) = 1$ if $x \in A$; $\mathbf{1}_A(x) = 0$ if $x \notin A$
$\mathcal{BV}(\Omega, [0,1])$	space of functions $f : \Omega \to [0,1]$ with bounded variation (cf. Definition 2.16)
$\mathcal{C}_c^k(\Omega, \mathbb{R}^n)$	set of all functions $f : \Omega \to \mathbb{R}^n$ being k-times continuously differentiable and with compact support, $k \in [1, \ldots, \infty]$
$\operatorname{diag}(\cdot)$	diagonal matrix, e.g. $\operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_n \end{bmatrix}$
$\operatorname{div}(\boldsymbol{p})$	divergence of vector field $\boldsymbol{p}$: $\operatorname{div}(\boldsymbol{p}) = \sum_{i=1}^{n} \frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{x}_i}$
$\operatorname{Gap}(\cdot)$	primal-dual gap, i.e. difference of primal and dual energies
$\mathcal{G}(\cdot)$	standard Gaussian distribution
$H(x)$	Heavyside step function, $H(x) = \mathbf{1}_{\{x<0\}}$
$\mathcal{H}^{n-1}(\cdot)$	$(n-1)$-dimensional Haussdorff measure
$\operatorname{int}(A), \operatorname{ext}(A)$	interior and exterior of a set A
I	image function $I : \Omega \to \mathbb{R}^d$ with d-dimensional pixel values
$L(\cdot)$	Lagrangian density
$\mathcal{L}^p(\Omega, \mathbb{R})$	Lebesgue space of functions with domain Ω, image $\mathbb{R}$ and finite p-norm
n	dimension of a function domain, e.g. $\mathbb{R}^n$
N	number of cameras in the scene
$\operatorname{Per}(A, \Omega)$	perimeter of set $A \subset \Omega$ in the domain Ω
$\operatorname{prox}_{\tau G}(u)$	proximity operator [58]
$\mathbb{R}$	set of real numbers
$\mathbb{R}_{\geq 0}$	set of positive real numbers $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$
S	silhouette (as a subset of the image domain $S \subset \Omega$)
$SO(3)$	special orthogonal group $SO(3) \subset \mathbb{R}^{3\times 3}$ (group of rotation matrices)
T	temporal domain $T \subset \mathbb{R}_{\geq 0}$
$\operatorname{TV}(u; \Omega)$	total variation of function u on the domain Ω
$\operatorname{TV}_g(u; \Omega)$	weighted total variation of function u on the domain Ω
u	implicit representation of a hypersurface
U	set of all implicit hypersurfaces
V	three dimensional volume domain $V \subset \mathbb{R}^3$
$V \times T$	spatio-temporal volume domain
V_t	target volume
$\boldsymbol{x}$	a point in 3D $\boldsymbol{x} \in V$ or 4D space $\boldsymbol{x} \in V \times T$ (depending on the context)
$\mathbb{Z}$	set of integer numbers
Σ	surface - being a manifold embedded in either $\mathbb{R}^3$ or $\mathbb{R}^4$
Ω	two dimensional image domain $\Omega \subset \mathbb{R}^2$
∂A	boundary of set A
π	orthogonal/perspective projection of points in 3D euclidean space
Π_A	orthogonal euclidean projection onto the set A

List of Abbreviations

1D, 2D, 3D, 4D	n-dimensional, $n = 1, 2, \ldots$
2.5D	3D points are parametrized via a 2D domain (depth map approach).
ACCV	Asian Conference on Computer Vision
ADMM	Alternating Direction Method of Multipliers (see [76])
BMVC	British Machine Vision Conference
BV	Bounded Variation
cf.	compare (from latin “confer”)
CUDA	Compute Unified Device Architecture (parallel computing platform by NVidia)
CPU	Central Processing Unit
CVPR	International Conference on Computer Vision and Pattern Recognition
DAGM	German Conference on Pattern Recognition (GCPR) formerly DAGM
e.g.	for example (from latin “exempli gratia”)
etc.	and so on (from latin “et cetera”)
ECCV	European Conference on Computer Vision
FISTA	Fast Iterative Shrinkage Algorithm (by Beck and Teboulle [20])
GCPR	German Conference on Pattern Recognition
GPU	Graphics Processing Unit
i.e.	that is (from latin “id est”)
ICCV	International Conference on Computer Vision
INRIA	Institut national de recherche en informatique et en automatique
LDFPI	Lagged Diffusivity Fixed Point Iteration (algorithm by Vogel and Oman [231])
MRF	Markov Random Field
NCC	Normalized Cross-Correlation
NURBS	Non-uniform rational B-spline
NVidia	Company which mainly produces graphics processors and related software.
PC	Personal Computer
PCA	Principal Component Analysis
PD	Primal-Dual
PDE	Partial Differential Equation
PMVS	Patch-based Multi-view Stereo (method by Furukawa et al. [91])
RAM	Random Access Memory
SfS	Shape from Shading
SOR	Successive Over-Relaxation
s.t.	subject to
TV	Total Variation
w/o	without

Own Publications

[1] Martin R. Oswald, Eno Töppe, Kalin Kolev, and Daniel Cremers. Non-parametric single view reconstruction of curved objects using convex optimization. In *Pattern Recognition (Proc. DAGM)*, Jena, Germany, September 2009. **received a DAGM Paper Award**.

[2] Daniel Cremers, Magnus Magnor, Martin R. Oswald, and Lihi Zelnik-Manor, editors. *Video Processing and Computational Video*, volume 7082 of *LNCS*. Springer, Berlin, Heidelberg, 2011.

[3] Eno Töppe, Martin R. Oswald, Daniel Cremers, and Carsten Rother. Image-based 3D modeling via cheeger sets. In *Proc. Asian Conference on Computer Vision (ACCV)*, Queenstown, New Zealand, November 2010. **received an Honorable Mention Award**.

[4] Eno Töppe, Martin R. Oswald, Daniel Cremers, and Carsten Rother. Silhouette-based variational methods for single view reconstruction. In D. Cremers, M. A. Magnor, M. R. Oswald, and L. Zelnik-Manor, editors, *Proceedings of the 2010 international conference on Video Processing and Computational Video*, pages 104–123, Berlin, Heidelberg, 2011. Springer-Verlag.

[5] Martin R. Oswald, Eno Töppe, and Daniel Cremers. Fast and globally optimal single view reconstruction of curved objects. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 534–541, Providence, Rhode Island, June 2012. video: http://youtu.be/59XooIf2zOM.

[6] Martin R. Oswald, Eno Töppe, Claudia Nieuwenhuis, and Daniel Cremers. A review of geometry recovery from a single image focusing on curved object reconstruction. In *Proceedings of the 2011 Conference on Innovations for Shape Analysis: Models and Algorithms*, Mathematics and Visualization, pages 343–378. Springer-Verlag, 2013.

[7] Martin R. Oswald and Daniel Cremers. A convex relaxation approach to space time multi-view 3D reconstruction. In *Internation Conference on Computer Vision (ICCV) - Workshop on Dynamic Shape Capture and Analysis (4DMOD)*, 2013. video: http://youtu.be/axGBJbawacA.

[8] Martin R. Oswald and Daniel Cremers. Surface normal integration for convex space-time multi-view reconstruction. In *Proc. of the British Machine and Vision Conference (BMVC)*, 2014. video: http://youtu.be/e9T4o0WHhPI.

[9] Martin R. Oswald, Jan Stühmer, and Daniel Cremers. Generalized connectivity constraints for spatio-temporal 3D reconstruction. In *Proc. European Conference on Computer Vision (ECCV)*, 2014. video: http://youtu.be/4H0GmCUDEsc.

[10] Tobias Gurdan, Martin R. Oswald, Daniel Gurdan, and Daniel Cremers. Spatial and temporal interpolation of multi-view image sequences. In *German Conference on Pattern Recognition (GCPR)*, Münster, Germany, September 2014. video: http://vimeo.com/104878209.

References

[11] Ehsan Aganj, Jean-Philippe Pons, and Renaud Keriven. Globally optimal spatio-temporal reconstruction from cluttered videos. In *Proc. Asian Conference on Computer Vision (ACCV)*, pages 667–678, 2009. (cited on page 97)

[12] Ehsan Aganj, Jean-Philippe Pons, Florent Ségonne, and Renaud Keriven. Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. (cited on page 97)

[13] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 2000. (cited on pages 12, 14, 17, and 130)

[14] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (second edition)*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2006. (cited on page 19)

[15] M. Aubry, K. Kolev, B. Goldluecke, and D. Cremers. Decoupling photometry and geometry in dense variational camera calibration. In *Proc. International Conference on Computer Vision (ICCV)*, 2011. (cited on page 145)

[16] S. Bae and F. Durand. Defocus magnification. In *Eurographics*, 2007. (cited on page 40)

[17] Roberto Bagnara. A unified proof for the convergence of Jacobi and Gauss-Seidel methods. *SIAM Review*, 37(1):pp. 93–97, 1995. (cited on page 27)

[18] Olga Barinova, Vadim Konushin, Anton Yakubenko, Keechang Lee, Hwasup Lim, and Anton Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *Proc. European Conference on Computer Vision (ECCV)*, pages 100–113, Berlin, Heidelberg, 2008. Springer-Verlag. (cited on page 46)

[19] Bruce Guenther Baumgart. *Geometric Modeling for Computer Vision*. PhD thesis, Stanford University, Stanford, CA, USA, 1974. AAI7506806. (cited on pages 95 and 96)

[20] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2:183–202, March 2009. (cited on pages 27, 72, and 148)

[21] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, Hélène Laurent, and Christophe Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19, July 2010. (cited on page 95)

[22] Ali Bigdelou, Alexander Ladikos, and Nassir Navab. Incremental visual hull reconstruction. In *Proc. of the British Machine and Vision Conference (BMVC)*, pages 1–11, 2009. (cited on page 97)

[23] Michael Bleyer, Carsten Rother, Pushmeet Kohli, Daniel Scharstein, and Sudipta Sinha. Object stereo – joint stereo matching and object segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3081–3088. IEEE, 2011. (cited on pages 126, 127, and 136)

[24] Mario Botsch and Leif Kobbelt. An intuitive framework for real-time freeform modeling. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, volume 23, pages 630–634, New

York, USA, August 2004. ACM Press/Addison-Wesley Publishing Co. (cited on page 48)

[25] Andrea Bottino, Luc Jaulin, and Aldo Laurentini. Reconstructing 3d objects from silhouettes with unknown viewpoints: The case of planar orthographic views. In *Progress in Pattern Recognition, Speech and Image Analysis, 8th Iberoamerican Congress on Pattern Recognition, CIARP*, Lecture Notes in Computer Science, pages 153–162. Springer, 2003. (cited on pages xvi, 95, and 96)

[26] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. (cited on pages 9, 19, 22, and 23)

[27] Edmond Boyer and Jean-Sébastien Franco. A hybrid approach for computing visual hulls of complex objects. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. (cited on page 96)

[28] Y. Boykov and M.-P. Jolly. Interactive organ segmentation using graph cuts. In *Medical Image Computing and Computer Assisted Interventions*, volume 1935 of *LNCS*, pages 276–286. Springer, 2000. (cited on page 50)

[29] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary region segmentation of objects in n-d images. In *Proc. International Conference on Computer Vision (ICCV)*, pages 105–112 vol.1, 2001. (cited on page 50)

[30] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. (cited on page 126)

[31] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. *Journal of Statistical Planning and Inference*, 37:28–47, 1986. (cited on pages xv and 62)

[32] Derek Bradley and Thabo Beeler. Local signal equalization for correspondence matching. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 1881–1887, 2013. (cited on page 144)

[33] Kristian Bredies and Dirk A. Lorenz. *Mathematische Bildverarbeitung - Einführung in Grundlagen und moderne Theorie*. Vieweg+Teubner, 2011. (cited on pages 11 and 19)

[34] Xavier Bresson, Selim Esedoḡlu, Pierre Vandergheynst, Jean-Philippe Thiran, and Stanley Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007. (cited on page 13)

[35] Adrian Broadhurst, Tom W. Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proc. International Conference on Computer Vision (ICCV)*, volume 1, pages 388–393. Department of Engineering,University of Cambridge,Cambridge, UK CB2 1PZ, 2001. (cited on page 96)

[36] C. G. Broyden. On convergence criteria for the method of successive over-relaxation (in Technical Notes and Short Papers). *Mathematics of Computation*, 18(85):136–141, January 1964. (cited on page 27)

[37] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Free-form mesh tracking: A patch-based approach. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1339–1346, 2010. (cited on page 99)

[38] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Probabilistic deformable surface tracking from multiple videos. In *Proc. European Conference on Computer Vision (ECCV)*, pages 326–339, 2010. (cited on page 99)

[39] Fatih Calakli and Gabriel Taubin. Ssd: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, 30(7):1993–2002, 2011. (cited on pages 97, 144, and 145)

[40] Jamylle Laurice Carter. *Dual Methods for Total Variation-Based Image Restoration*. PhD thesis, University of California, Los Angeles, 2001. (cited on page 13)

[41] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997. (cited on pages 15 and 18)

[42] Vicent Caselles, Ron Kimmel, Guillermo Sapiro, and Catalina Sbert. Three dimensional object modeling via minimal surfaces. In *Proc. European Conference on Computer Vision (ECCV)*, pages 97–106, 1996. (cited on page 18)

[43] Vicent Caselles, Ron Kimmel, Guillermo Sapiro, and Catalina Sbert. Minimal surfaces based object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):394–398, 1997. (cited on page 18)

[44] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock. An introduction to total variation for image analysis. In *In Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter*, 2010. (cited on pages 11 and 13)

[45] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004. (cited on page 32)

[46] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011. (cited on pages 28 and 29)

[47] T. Chan, S. Esedoḡlu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006. (cited on pages 18, 20, 61, and 63)

[48] Tony F. Chan, Gene H. Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977, May 1999. (cited on page 13)

[49] Tony F. Chan and Pep Mulet. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Applied Mathematics*, 36(2):354–367, February 1999. (cited on page 26)

[50] Ju Yong Chang, Kyoung Mu Lee, and Sang Uk Lee. Multiview normal field integration using level set methods. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. (cited on page 114)

[51] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. International Conference on Image Processing (ICIP)*, pages 168–172, 1994. (cited on page 72)

[52] J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Problems in analysis*. Princeton Univ. Press, Princeton, N.J., 1970. (cited on pages 59 and 64)

[53] Chao Chen, Daniel Freedman, and Christoph H. Lampert. Enforcing topological constraints in random field image segmentation. In *CVPR*, pages 2089–2096, 2011. (cited on page 126)

[54] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, 32(6):195:1–195:10, November 2013. (cited on page 44)

[55] Yu Chen and Roberto Cipolla. Single and sparse view 3d reconstruction by learning shape priors. *Computer Vision and Image Understanding*, 115:586–602, May 2011. (cited on pages 41, 45, 46, 47, and 48)

[56] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and

balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993. (cited on pages 43 and 59)

[57] Carlo Colombo, Alberto Del Bimbo, Alberto Del, and Federico Pernici. Metric 3d reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:99–114, 2005. (cited on pages 44 and 47)

[58] Patrick Louis Combettes and Jean-Christophe Pesquet. Proximal Splitting Methods in Signal Processing. In R.S.; Combettes P.L.; Elser V.; Luke D.R.; Wolkowicz H. (Eds.) Bauschke, H.H.; Burachik, editor, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011. (cited on pages 28, 29, and 147)

[59] Paul Concus. Numerical solution of the minimal surface equation. *Mathematics of Computation*, 21:340–350, 1967. (cited on page 72)

[60] Jérôme Courchay, Jean-Philippe Pons, Pascal Monasse, and Renaud Keriven. Dense and accurate spatio-temporal multi-view stereovision. In *Proc. Asian Conference on Computer Vision (ACCV)*, pages 11–22, 2009. (cited on page 98)

[61] F. Courteille, A. Crouzil, J.-D. Durou, and P. Gurdjos. Towards shape from shading under realistic photographic conditions. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 277–280, Cambridge, UK, August 2004. IEEE Computer Society. (cited on page 39)

[62] D. Cremers and K. Kolev. Multiview stereo and silhouette consistency via convex functionals over convex domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1161–1174, 2011. (cited on pages 115 and 126)

[63] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. (cited on pages 41, 44, 46, 47, and 48)

[64] Daniel E. Crispell, Joseph L. Mundy, and Gabriel Taubin. A variable-resolution probabilistic three-dimensional model for change detection. *IEEE T. Geoscience and Remote Sensing*, 50(2):489–500, 2012. (cited on pages 97 and 145)

[65] B. Dacorogna. *Introduction to the Calculus of Variations*. Imperial College Press, 2009. (cited on pages 19 and 70)

[66] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004. (cited on page 28)

[67] M. Daum and G. Dudek. On 3-d surface reconstruction using shape from shadows. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 461–468. IEEE Computer Society, 1998. (cited on page 39)

[68] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 98:1–98:10, New York, NY, USA, 2008. ACM. (cited on page 99)

[69] Edilson de Aguiar, Christian Theobalt, Carsten Stoll, and Hans-Peter Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. (cited on page 99)

[70] Erick Delage, Honglak Lee, and Andrew Y. Ng. Automatic single-image 3d reconstructions of indoor Manhattan world scenes. In Sebastian Thrun, Rodney A. Brooks, and Hugh F. Durrant-Whyte, editors, *Proc. of the International Symposium of Robotics Research*, pages 305–321, San Francisco, CA, USA, October 2005. Springer Tracts in

Advanced Robotics. (cited on pages 39, 40, 41, 44, 46, and 47)

[71] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1486–1493, 2014. (cited on page 145)

[72] Tamal K. Dey, Fengtao Fan, and Yusu Wang. An efficient computation of handle and tunnel loops via reeb graphs. *ACM Trans. Graph.*, 32(4):32, 2013. (cited on page 131)

[73] Tamal K. Dey, Kuiyu Li, Jian Sun, and David Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. *ACM Trans. Graph.*, 27(3), 2008. (cited on pages 130 and 131)

[74] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions of Several Variables, Oberwolfach 1976*, volume 571, pages 85–100. Springer, 1977. (cited on page 42)

[75] J.-D. Durou, M. Falcone, and M. Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 109:22–43, January 2008. (cited on page 39)

[76] Jonathan Eckstein and Dimitri P. Bertsekas. On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3):293–318, June 1992. (cited on pages 24 and 148)

[77] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, December 2004. (cited on pages 104, 115, 117, and 124)

[78] Lawrence C. Evans. *Partial differential equations*. Graduate studies in mathematics. American Mathematical Society, Providence (R.I.), 1998. RÃľimpr. avec corrections : 1999, 2002. (cited on page 24)

[79] Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7:336–344, 1999. (cited on pages 18, 94, and 116)

[80] H. Federer. *Geometric measure theory*. Grundlehren der mathematischen Wissenschaften. Springer, 1969. (cited on page 17)

[81] Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):pp. 418–491, December 1959. (cited on page 17)

[82] Wendell H. Fleming and Raymond Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, 1960. (cited on page 17)

[83] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice (2nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990. (cited on page 18)

[84] Jean-Sébastien Franco and Edmond Boyer. Exact polyhedral visual hulls. In *Proc. of the British Machine and Vision Conference (BMVC)*, pages 1–10, 2003. (cited on page 96)

[85] Jean-Sébastien Franco and Edmond Boyer. Fusion of multi-view silhouette cues using a space occupancy grid. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1747–1753, 2005. (cited on page 96)

[86] Jean-Sébastien Franco and Edmond Boyer. Efficient polyhedral modeling from silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):414–427, 2009. (cited on page 96)

[87] A.R.J. Francois and G.G. Medioni. Interactive 3d model extraction from a single image.

Image and Vision Computing, 19(6):317–328, April 2001. (cited on page 43)

[88] Simon Fuhrmann and Michael Goesele. Fusion of depth maps with multiple scales. *ACM Trans. Graph.*, 30(6):148, 2011. (cited on page 144)

[89] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. (cited on pages 96 and 99)

[90] Yasutaka Furukawa and Jean Ponce. Dense 3d motion capture from synchronized video streams. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. (cited on page 99)

[91] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, August 2010. (cited on pages 96, 114, 115, 121, 122, 123, 124, 135, 137, and 148)

[92] I.M. Gelfand, S.V. Fomin, and R.A. Silverman. *Calculus of Variations*. Dover Books on Mathematics. Dover Publications, 2000. (cited on page 19)

[93] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. (cited on pages 96 and 114)

[94] B. Goldluecke, I. Ihrke, C. Linz, and M. Magnor. Weighted minimal hypersurface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1194–1208, July 2007. (cited on pages 18 and 98)

[95] B. Goldluecke and M. Magnor. Space-time isosurface evolution for temporally coherent 3D reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume I, pages 350–355, July 2004. (cited on pages 18 and 98)

[96] B. Goldluecke, E. Strekalovskiy, and D. Cremers. Tight convex relaxations for vector-valued labeling. *SIAM Journal on Imaging Sciences*, 6(3):1626–1664, 2013. (cited on page 20)

[97] Bastian Goldluecke. *Multi-Camera Reconstruction and Rendering for Free-Viewpoint Video*. PhD thesis, Universität des Saarlandes / Max-Planck-Institut für Informatik, Saarbrücken, Germany, 2006. (cited on page 98)

[98] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM J. Img. Sci.*, 2(2):323–343, April 2009. (cited on page 24)

[99] Li Guan, Jean-Sébastien Franco, Edmond Boyer, and Marc Pollefeys. Probabilistic 3d occupancy flow with latent silhouette cues. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1379–1386, 2010. (cited on page 98)

[100] Li Guan, Jean-Sébastien Franco, and Marc Pollefeys. Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction. *International Journal of Computer Vision*, 90(3):283–303, 2010. (cited on page 97)

[101] Li Guan, Sudipta N. Sinha, Jean-Sébastien Franco, and Marc Pollefeys. Visual hull construction in the presence of partial occlusion. In *3rd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2006), 14-16 June 2006, Chapel Hill, North Carolina, USA*, pages 413–420, 2006. (cited on page 96)

[102] J.-Y. Guillemaut and A. Hilton. Space-time joint multi-layer segmentation and depth estimation. In *Proc. International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 440–447, 2012. (cited on page 97)

[103] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisser-

man. Geodesic star convexity for interactive image segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3129–3136. IEEE, 2010. (cited on page 127)

[104] Feng Han and Song-Chun Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *Proceedings of the First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, pages 12–20, Washington, DC, USA, 2003. IEEE Computer Society. (cited on pages 39, 41, 45, 46, and 47)

[105] Xiao Han, Chenyang Xu, and Jerry L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003. (cited on page 126)

[106] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, 2013. (cited on pages 18, 97, and 144)

[107] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. (cited on page 96)

[108] T. Hassner and R. Basri. Example based 3d reconstruction from single 2d images. In *Beyond Patches Workshop at IEEE Conference on Computer Vision and Pattern Recognition*, page 15. IEEE Computer Society, June 2006. (cited on pages 40, 45, and 47)

[109] Michael Hatzitheodorou. The derivation of 3-d surface shape from shadows. In *Proceedings of a workshop on Image understanding*, pages 1012–1020, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. (cited on page 39)

[110] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*, 24(3):577–584, 2005. (cited on pages 39, 40, 41, 45, 46, and 47)

[111] Wei Hong, Allen Yang Yang, Kun Huang, and Yi Ma. On symmetry and multiple-view geometry: Structure, pose, and calibration from a single image. *International Journal of Computer Vision*, 60:241–265, 2004. (cited on pages 41 and 45)

[112] Radu P. Horaud and Michael Brady. On the geometric interpretation of image contours. *Artificial Intelligence*, 37(1-3):333–353, December 1988. Special Issue on Geometric Reasoning. (cited on page 39)

[113] Alexander Hornung and Leif Kobbelt. Robust and efficient photo-consistency estimation for volumetric 3d reconstruction. In *Proc. European Conference on Computer Vision (ECCV)*, pages 179–190, 2006. (cited on page 144)

[114] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. (cited on page 46)

[115] Takeo Igarashi. Adaptive unwrapping for interactive texture painting. In *ACM Symposium on Interactive 3D Graphics*, pages 209–216, New York, NY, USA, 2001. ACM. (cited on page 82)

[116] Takeo Igarashi. Smooth meshes for sketch-based freeform modeling. In *Proceedings of the 2003 Symposium on Interactive 3D graphics*, pages 139–142, New York, NY, USA, 2003. ACM Press. (cited on page 82)

[117] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. (cited

on pages xvi, 39, 48, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, and 141)

[118] William H. Meeks III and Joaquín Pérez. The classical theory of minimal surfaces. *Bulletin of the American Mathematical Society*, 48:325–407, 2011. (cited on pages 18 and 70)

[119] William H. Meeks III and Joaquín Pérez. *A Survey on Classical Minimal Surface Theory*. University lecture series. American Mathematical Society, 2012. (cited on page 15)

[120] K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–185, 1981. (cited on page 39)

[121] Institut national de recherche en informatique et en automatique (INRIA) Rhône Alpes. 4d repository. `http://4drepository.inrialpes.fr/` (visited: Aug 22, 2014). (cited on pages xvi, 93, 107, 113, 121, 122, 124, 125, and 135)

[122] Michal Jancosek and Tomás Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3121–3128, 2011. (cited on pages 96, 101, 108, 109, 111, 121, 122, 123, 124, 135, and 137)

[123] P. Joshi and N. Carr. Repoussé: Automatic inflation of 2d art. In *Proceedings of the sixth Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 49ÂŰ–56, Aire-la-Ville, Switzerland, 2008. Eurographics Association. (cited on page 48)

[124] W.M. Kahan. *Gauss-Seidel Methods of solving large systems of linear equations*. PhD thesis, Toronto, Canada, University of Toronto, 1958. (cited on page 27)

[125] Takeo Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*, 17:409 – 460, 1981. (cited on page 44)

[126] Olga A. Karpenko and John F. Hughes. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Transactions on Graphics*, 25/3:589–598, 2006. (cited on pages 39 and 48)

[127] Olga A. Karpenko, John F. Hughes, and Ramesh Raskar. Free-form sketching with variational implicit surfaces. *Computer Graphics Forum*, 21(3):585–594, 2002. (cited on page 39)

[128] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. (cited on page 18)

[129] Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 61–70, 2006. (cited on pages 96, 114, 121, 122, 123, 124, 135, 137, and 144)

[130] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29, 2013. (cited on pages 114 and 144)

[131] J. Kender and E. Smith. Shape from darkness. In *Proc. International Conference on Computer Vision (ICCV)*, pages 539–546, 1987. (cited on page 39)

[132] Satyanad Kichenassamy, Arun Kumar, Peter J. Olver, Allen Tannenbaum, and Anthony J. Yezzi. Gradient flows and geometric active contour models. In *Proc. International Conference on Computer Vision (ICCV)*, pages 810–815, 1995. (cited on page 18)

[133] Maria Klodt, Thomas Schoenemann, Kalin Kolev, Marek Schikora, and Daniel Cremers. An experimental comparison of discrete and continuous shape optimization methods. In *Proc. European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008. (cited on page 32)

[134] Kalin Kolev. *Convexity in Image-Based 3D Surface Reconstruction*. PhD thesis, Tech-

nische Universität München, Munich, Germany, 2011. (cited on pages 6, 18, 26, 93, 97, and 141)

[135] Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. (cited on pages xvi, 18, 26, 54, 56, 97, 101, 102, 105, 107, 108, 115, 118, and 126)

[136] Kalin Kolev, Thomas Pock, and Daniel Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In *Proc. European Conference on Computer Vision (ECCV)*, Heraklion, Greece, September 2010. (cited on pages 97, 114, 115, and 116)

[137] Kevin Köser, Christopher Zach, and Marc Pollefeys. Dense 3d reconstruction of symmetric scenes from a single image. In *Pattern Recognition - 33rd DAGM Symposium, Frankfurt/Main, Germany, August 31 - September 2, 2011. Proceedings*, pages 266–275, 2011. (cited on page 143)

[138] Panagiotis Koutsourakis, Loic Simon, Olivier Teboul, Georgios Tziritas, and Nikos Paragios. Single view reconstruction using shape grammars for urban environments. In *Proc. International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009. IEEE. (cited on pages 39, 41, 44, and 47)

[139] Akash M Kushal, Subhajit Sanyal, Vikas Bansal, and Subhashis Banerjee. A simple method for interactive 3d reconstruction and camera calibration from a single view. In *In Proceedings Indian Conference in Computer Vision, Graphics and Image Processing*, 2002. (cited on page 44)

[140] Avanish Kushal and Steven M. Seitz. Single view reconstruction of piecewise swept surfaces. In *International Conference on 3D Vision (3DV)*, pages 239–246, 2013. (cited on pages 41 and 44)

[141] Kiriakos N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *Proc. International Conference on Computer Vision (ICCV)*, pages 307 – 314, 1999. 20-27 Sept., Volume 1. (cited on page 96)

[142] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. *Comput. Graph. Forum*, 28(8):2275–2290, 2009. (cited on page 96)

[143] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 89–96, 2014. (cited on page 143)

[144] Lubor Ladicky, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 468–484, 2014. (cited on page 143)

[145] Alexander Ladikos, Selim Benhimane, and Nassir Navab. Multi-view reconstruction using narrow-band graph-cuts and surface normal optimization. In *Proc. of the British Machine and Vision Conference (BMVC)*, pages 1–10, 2008. (cited on pages 114 and 118)

[146] Alexander Ladikos and Nassir Navab. Real-time 3d reconstruction for occlusion-aware interactions in mixed reality. In *Advances in Visual Computing, 5th International Symposium, ISVC 2009, Las Vegas, NV, USA, November 30 - December 2, 2009, Proceedings, Part I*, pages 480–489, 2009. (cited on page 97)

[147] Joseph Louis Lagrange. Essai d'une nouvelle methode pour determiner les maxima et les minima des formules integrales indefinies. *Miscellanea Taurinensia 2, Ouvres*, 1:335–362, 1760. (cited on pages 18 and 70)

[148] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. In

IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 150 – 162, 1994. Feb., Volume 16, Issue 2. (cited on pages 95 and 96)

[149] Anat Levin. Analyzing depth from coded aperture sets. In *Proc. European Conference on Computer Vision (ECCV)*, pages 214–227. Springer-Verlag, 2010. (cited on page 40)

[150] Stan Z. Li. *Markov random field modeling in computer vision.* Computer science workbench. Springer, London, UK, 1995. (cited on page 44)

[151] Yunfeng Li, Zygmunt Pizlo, and Robert M. Steinman. A computational model that recovers the 3d shape of an object from a single 2d retinal representation. *Vision Research*, 49:979 – 991, May 2009. (cited on pages 5, 39, 41, and 44)

[152] Zhenguo Li, Jianzhuang Liu, and Xiaoou Tang. A closed-form solution to 3d reconstruction of piecewise planar objects from single images. *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007. (cited on page 45)

[153] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Proc. EuroGraphics*, volume 18, pages 39–50, 1999. (cited on page 44)

[154] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1253–1260, 2010. (cited on page 40)

[155] Yebin Liu, Qionghai Dai, and Wenli Xu. A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):407–418, May 2010. (cited on pages 107, 108, and 109)

[156] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169, August 1987. (cited on pages 107 and 120)

[157] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987. (cited on page 40)

[158] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23(2):149–168, 1997. (cited on page 40)

[159] T. Möllenhoff, C. Nieuwenhuis, E. Töppe, and D. Cremers. Efficient convex optimization for minimal partition problems with volume constraints. In *Proceedings of the International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2013. (cited on page 24)

[160] Julian Musielak and Wladyslaw Orlicz. On generalized variations (I). *Studia Mathematica*, 18(1):11–41, 1959. (cited on page 13)

[161] T. Nagai, T. Naruse, M. Ikehara, and A. Kurematsu. Hmm-based surface reconstruction from single images. *Proc. International Conference on Image Processing (ICIP)*, 2:II–561–II–564 vol.2, 2002. (cited on page 45)

[162] S. K. Nayar, K. Ikeuchi, and T. Kanade. Shape from interreflections. *Proc. International Conference on Computer Vision (ICCV)*, pages 2–11, July 1990. (cited on page 39)

[163] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph.*, 26(3):41, 2007. (cited on pages 39 and 48)

[164] Jan Neumann and Yiannis Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision*, 47(1-3):181–193, 2002. (cited on page 98)

[165] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013. (cited on page 145)

[166] Claudia Nieuwenhuis, Eno Töppe, Lena Gorelick, Olga Veksler, and Yuri Boykov. Efficient squared curvature. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4098–4105, 2014. (cited on page 144)

[167] Sebastian Nowozin and Christoph H Lampert. Global connectivity potentials for random field models. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 818–825. IEEE, 2009. (cited on page 127)

[168] Carl Olsson, Martin Byröd, Niels Chr. Overgaard, and Fredrik Kahl. Extending continuous cuts: Anisotropic metrics and expansion moves. In *Proc. International Conference on Computer Vision (ICCV)*, pages 405–412. IEEE, 2009. (cited on page 115)

[169] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, pages 12–49, 1988. (cited on page 18)

[170] Pietro Perona and Jitendra Malik. Scale-space and edge-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. (cited on page 71)

[171] Massimo Piccardi. Background subtraction techniques: a review. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics: The Hague, Netherlands, 10-13 October 2004*, pages 3099–3104, 2004. (cited on page 95)

[172] Zygmunt Pizlo, Tadamasa Sawada, Yunfeng Li, Walter G. Kropatsch, and Robert M. Steinman. New approach to the perception of 3d shape based on veridicality, complexity, symmetry and volume. *Vision Research*, 50:1–11, January 2010. (cited on page 5)

[173] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the piecewise smooth mumford-shah functional. In *Proc. International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009. (cited on pages 28, 29, and 61)

[174] Thomas Pock. *Fast Total Variation for Computer Vision*. PhD thesis, University of Graz, Austria, January 2008. (cited on page 13)

[175] Thomas Pock and Antonin Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1762–1769, Washington, DC, USA, 2011. (cited on pages 29, 30, 106, 119, 134, and 135)

[176] Thomas Pollard and Joseph L. Mundy. Change detection in a 3-d world. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1 – 6, Minneapolis, USA, 2007. IEEE. (cited on pages 97 and 105)

[177] Jean-Philippe Pons, Renaud Keriven, and Olivier D. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 822–827, 2005. (cited on page 98)

[178] Jean-Philippe Pons, Renaud Keriven, and Olivier D. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007. (cited on page 98)

[179] Jean-Philippe Pons, Renaud Keriven, Olivier D. Faugeras, and Gerardo Hermosillo. Variational stereovision and 3d scene flow estimation with statistical similarity measures. In *Proc. International Conference on Computer Vision (ICCV)*, pages 597–602, 2003. (cited on page 98)

[180] E. Prados and O. Faugeras. Shape from shading: a well-posed problem? In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 870–877, San Diego, California, Etats-Unis, June 2005. IEEE. (cited on page 39)

[181] M. Prasad. *Class-based Single View Reconstruction*. PhD thesis, University of Oxford, July 2009. (cited on pages 74, 77, 81, 82, 83, 84, 85, and 87)

[182] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Fast and controllable 3D modelling from silhouettes. In *Eurographics, Short Papers*, pages 9–12, Dublin, Irland, September 2005. (cited on pages 39, 43, and 82)

[183] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1354, 2006. (cited on pages xvi, 39, 40, 43, 46, 47, 48, 65, 74, 75, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, and 141)

[184] Srikumar Ramalingam and Matthew Brand. Lifting 3d manhattan lines from a single image. In *Proc. International Conference on Computer Vision (ICCV)*, 2013. (cited on page 44)

[185] Christian Reinbacher, Thomas Pock, Christian Bauer, and Horst Bischof. Variational segmentation of elongated volumetric structures. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. (cited on pages 115, 116, and 118)

[186] Christian Richardt, Carsten Stoll, Neil A. Dodgson, Hans-Peter Seidel, and Christian Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2), May 2012. (cited on page 98)

[187] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970. (cited on pages 9, 11, 19, and 23)

[188] R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften*. Springer, 1998. (cited on pages 9, 11, and 19)

[189] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. (cited on pages 50 and 95)

[190] Diego Rother and Guillermo Sapiro. Seeing 3d objects in a single 2d image. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1819–1826, 2009. (cited on pages 41, 45, 46, and 47)

[191] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, November 1992. (cited on page 24)

[192] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76, 2007. (cited on page 39)

[193] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. (cited on pages 39, 40, 45, 46, and 47)

[194] T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *International Journal of Computer Vision*, 99:53–68, 2012. (cited on page 144)

[195] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*,

pages 519–528, Washington, DC, USA, 2006. IEEE Computer Society. (cited on page 96)

[196] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999. (cited on page 96)

[197] Laura Sevilla-Lara, Deqing Sun, Erik G. Learned-Miller, and Michael J. Black. Optical flow estimation with channel constancy. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 423–438, 2014. (cited on page 144)

[198] Andrei Sharf, Dan A. Alcantara, Thomas Lewiner, Chen Greif, Alla Sheffer, Nina Amenta, and Daniel Cohen-Or. Space-time surface reconstruction using incompressible flow. In *ACM SIGGRAPH Asia 2008 papers*, pages 110:1–110:10, New York, NY, USA, 2008. ACM. (cited on page 98)

[199] soapbubble.dk. Picture of a catenoid-shaped soap bubble. Licensed under the Creative Commons Attribution-NonCommercial 3.0 License. http://soapbubble.dk/english/science/the-geometry-of-soap-films-and-soap-bubbles/ (visited: Sep 25, 2014). (cited on page 15)

[200] Nir Sochen, Ron Kimmel, and Ravi Malladi. A general framework for low level vision. *IEEE Transactions on Image Processing*, 7:310–318, 1997. (cited on page 72)

[201] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007. (cited on page 98)

[202] F. Steinbruecker, C. Kerl, J. Sturm, and D. Cremers. Large-scale multi-resolution surface reconstruction from RGB-D sequences. In *Proc. International Conference on Computer Vision (ICCV)*, Sydney, Australia, 2013. (cited on page 144)

[203] F. Steinbruecker, J. Sturm, and D. Cremers. Volumetric 3d mapping in real-time on a CPU. In *Proc. International Conference on Robotics and Automation (ICRA)*, Hongkong, China, 2014. (cited on page 144)

[204] Gilbert Strang. Maximal flow through a domain. *Mathematical Programming*, 26(2):123–143, June 1983. (cited on page 20)

[205] J. Stühmer, P. Schröder, and D. Cremers. Tree shape priors with connectivity constraints using convex relaxation on general graphs. In *Proc. International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013. (cited on pages 125, 126, 127, 128, 129, 130, 136, and 137)

[206] Peter F Sturm and Stephen J Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *Proc. of the British Machine and Vision Conference (BMVC)*, pages 265–274. British Machine Vision Association, 1999. (cited on page 44)

[207] Jian Sun, Weiwei Zhang, Xiaoou Tang, and Heung-Yeung Shum. Background cut. In *Proc. European Conference on Computer Vision (ECCV)*, pages 628–641, 2006. (cited on page 95)

[208] Boaz J. Super and Alan C. Bovik. Shape from texture using local spectral moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:333–343, April 1995. (cited on page 40)

[209] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision*, 1:211–221, 1987. (cited on pages 43 and 44)

[210] E. Töppe, C. Nieuwenhuis, and D. Cremers. Relative volume constraints for single view

reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, 2013. (cited on page 143)

[211] Eno Töppe. *Convex Optimization Methods for Single View 3D Reconstruction*. PhD thesis, Technische Universität München, Munich, Germany, June 2013. (cited on page 7)

[212] J.L. Troutman. *Variational Calculus and Optimal Control: Optimization With Elementary Convexity*. Undergraduate Texts in Mathematics. Springer Verlag, 2012. (cited on page 19)

[213] Tony Tung, Shohei Nobuhara, and Takashi Matsuyama. Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1709–1716, 2009. (cited on page 98)

[214] Faith Ulupinar and Ramakant Nevatia. Shape from contour: Straight homogeneous generalized cylinders and constant cross section generalized cylinders. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:120–135, February 1995. (cited on page 39)

[215] Ali Osman Ulusoy, Octavian Biris, and Joseph L. Mundy. Dynamic probabilistic volumetric models. In *Proc. International Conference on Computer Vision (ICCV)*, pages 505–512, 2013. (cited on pages 97 and 145)

[216] Benjamin Ummenhofer and Thomas Brox. Dense 3d reconstruction with a hand-held camera. In *DAGM/OAGM Symposium'12*, pages 103–112, 2012. (cited on pages 18 and 97)

[217] Gozde B. Unal, Anthony J. Yezzi, Stefano Soatto, and Gregory G. Slabaugh. A variational approach to problems in calibration of multiple cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1322–1338, 2007. (cited on page 145)

[218] Markus Unger, Thomas Mauthner, Thomas Pock, and Horst Bischof. Tracking as segmentation of spatial-temporal volumes by anisotropic weighted tv. In *Proceedings of the International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 193–206, Berlin, Heidelberg, 2009. Springer-Verlag. (cited on pages 101, 102, and 106)

[219] Markus Unger, Thomas Pock, Daniel Cremers, and Horst Bischof. TVSeg - interactive total variation based image segmentation. In *Proc. of the British Machine and Vision Conference (BMVC)*, Leeds, UK, September 2008. (cited on pages 50 and 95)

[220] Markus Unger, Thomas Pock, Manuel Werlberger, and Horst Bischof. A convex approach for variational super-resolution. In *Pattern Recognition - 32nd DAGM Symposium, Darmstadt, Germany, September 22-24, 2010. Proceedings*, pages 313–322, 2010. (cited on page 5)

[221] S. Utcke and A. Zisserman. Projective reconstruction of surfaces of revolution. In *Pattern Recognition (Proc. DAGM)*, pages 93–102, 2003. (cited on page 44)

[222] Kiran Varanasi, Andrei Zaharescu, Edmond Boyer, and Radu Horaud. Temporal surface tracking using mesh evolution. In *Proc. European Conference on Computer Vision (ECCV)*, pages 30–43, 2008. (cited on page 99)

[223] Sundar Vedula, Simon Baker, and Takeo Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. *ACM Trans. Graph.*, 24(2):240–261, 2005. (cited on page 98)

[224] Sundar Vedula, Simon Baker, Peter Rander, Robert T. Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proc. International Conference on Computer Vision (ICCV)*, pages 722–729, 1999. (cited on page 98)

[225] Sundar Vedula, Simon Baker, Steven M. Seitz, and Takeo Kanade. Shape and motion carving in 6d. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2592–2598, 2000. (cited on page 98)

[226] Thomas Vetter. Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28, June 1998. (cited on pages 40 and 45)

[227] Sara Vicente and Lourdes de Agapito. Balloon shapes: Reconstructing and deforming objects with volume from images. In *International Conference on 3D Vision (3DV), Seattle, Washington, USA, June 29 - July 1*, pages 223–230, 2013. (cited on page 143)

[228] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. (cited on pages 126 and 128)

[229] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3), 2008. (cited on page 99)

[230] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul E. Debevec, Jovan Popovic, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graph.*, 28(5), 2009. (cited on page 114)

[231] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Applied Mathematics*, 17(1):227–238, 1996. (cited on pages 24, 26, and 148)

[232] C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Transactions on Image Processing*, 7:813–824, 1998. (cited on pages 26 and 72)

[233] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a rigid motion prior. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1291–1298, 2011. (cited on page 98)

[234] H-H. Vu, R. Keriven, P. Labatut, and J.-P Pons. Towards high-resolution large-scale multi-view stereo. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Jun 2009. (cited on pages 96, 114, and 118)

[235] Guohui Wang, Wei Su, and Yugui Song. A new shape from shading approach for specular surfaces. In *Proceedings of the Third international conference on Artificial intelligence and computational intelligence - Volume Part III*, Lecture Notes in Computer Science, pages 71–78, Berlin, Heidelberg, 2011. Springer-Verlag. (cited on page 39)

[236] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, 95(1):29–51, 2011. (cited on page 98)

[237] Michael Weinmann, Aljosa Osep, Roland Ruiters, and Reinhard Klein. Multi-view normal field integration for 3d reconstruction of mirroring objects. *Proceedings of the International Conference on Computer Vision*, pages 2504–2511, December 2013. (cited on page 114)

[238] William Welch and Andrew P. Witkin. Free-form shape design using triangulated surfaces. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 247–256, 1994. (cited on page 48)

[239] T. Whelan, M. Kaess, M.F. Fallon, H. Johannsson, J.J. Leonard, and J.B. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012. (cited on page 145)

[240] Lance Williams. 3d paint. *Proceedings of the 1990 symposium on Interactive 3D graph-*

ics, Snowbird, Utah, United States, pages 225–233, February 1990. (cited on page 48)

[241] Kwan-Yee K. Wong, Paulo R. S. Mendonça, and Roberto Cipolla. Reconstruction of surfaces of revolution from single uncalibrated views. In *Proc. of the British Machine and Vision Conference (BMVC)*, pages 265–272, Cardiff, 2002. (cited on page 44)

[242] Chenglei Wu, Kiran Varanasi, Yebin Liu, Hans-Peter Seidel, and Christian Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1108–1115, 2011. (cited on page 114)

[243] Stephan Würmlin, Edouard Lamboray, and Markus H. Gross. 3d video fragments: dynamic point samples for real-time free-viewpoint video. *Computers & Graphics*, 28(1):3–14, 2004. (cited on page 97)

[244] Anthony J. Yezzi and Stefano Soatto. Stereoscopic segmentation. In *Proc. International Conference on Computer Vision (ICCV)*, pages 59–66, 2001. (cited on page 18)

[245] David M. Young, Jr. *Iterative Methods for Solving Partial Difference Equations of Elliptic Type*. PhD thesis, Harvard University, Cambridge, Mass, 1951. (cited on page 26)

[246] Y. Yu and J. Chang. Shadow graphs and surface reconstruction. In *Proc. European Conference on Computer Vision (ECCV)*, pages 31–45, 2002. (cited on page 39)

[247] Xenophon Zabulis and Kostas Daniilidis. Multi-camera reconstruction based on surface normal estimation and best viewpoint selection. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004), 6-9 September 2004, Thessaloniki, Greece*, pages 733–740. IEEE Computer Society, 2004. (cited on page 114)

[248] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. (cited on pages 101, 105, and 117)

[249] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: An interface for sketching 3d scenes. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 163–170, New York, NY, USA, 1996. ACM. (cited on page 48)

[250] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. A memory-efficient kinectfusion using octree. In Shi-Min Hu and Ralph R. Martin, editors, *Computational Visual Media*, volume 7633 of *Lecture Notes in Computer Science*, pages 234–241. Springer Berlin Heidelberg, 2012. (cited on page 144)

[251] Yun Zeng, Dimitris Samaras, Wei Chen, and Qunsheng Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n–d images. *Computer Vision and Image Understanding*, 112(1):81–90, 2008. (cited on page 126)

[252] Li Zhang, Brian Curless, and Steven M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 367–374, June 2003. (cited on page 97)

[253] Li Zhang, Guillaume Dugas-Phocion, Jean-Sebastien Samson, and Steven M. Seitz. Single view modeling of free-form scenes. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 990–997, December 2001. (cited on pages xvi, 40, 42, 43, 47, 48, 74, 77, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, and 141)

[254] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999. (cited on page 39)

[255] Mingqiang Zhu. *Fast Numerical Algorithms for Total Variation Based Image Restoration*. PhD thesis, University of California, Los Angeles, 2008. (cited on page 13)