TECHNISCHE UNIVERSITÄT MÜNCHEN

FAKULTÄT FÜR INFORMATIK
COMPUTER VISION GROUP

Convex Relaxation Methods
for Image Segmentation and Stereo Reconstruction

Maria Klodt

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. Nils Thuerey

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Daniel Cremers
2. Univ.-Prof. Dr. Dr. Wolfgang Förstner
   Rheinische Friedrich-Wilhelms-Universität Bonn (schriftliche Beurteilung)
2. Univ.-Prof. Dr. Bjoern Menze (mündliche Prüfung)

Die Dissertation wurde am 06.10.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 01.12.2014 angenommen.

Abstract

The thesis presents convex relaxation methods for image segmentation and stereo reconstruction. Image segmentation aims at segmenting an image domain to meaningful regions which ideally correspond to the different objects depicted in the scene. Stereo reconstruction aims at inferring 3D structures from a set of images depicting a scene. Image segmentation and stereo reconstruction belong to the most studied problems in computer vision, with diverse applications ranging from medical image analysis to autonomous navigation of mobile robots and automated interpretation of aerial images. Convex relaxation methods have been established as a powerful technique for finding globally optimal solutions for numerous computer vision problems. Convex formulations allow for finding these solutions independent of initializations.

A contribution of the thesis is a convex formulation of image segmentation with moment constraints. The lower order constraints include the area, centroid and covariance dimensions of a shape. Orthogonal projections onto the respective constraint sets allow for efficient optimization. A quantitative evaluation on a set of medical images has shown that the average segmentation error can be reduced from 12% to 0.35%. It was further shown that the moment constraints can make object tracking in image sequences more robust, especially in the case of highly similar color distributions in foreground and background regions. An extension to object tracking in RGB-D images allows for scale-aware tracking in the absolute 3D space rather than the projected image plane. This allows to impose shape constraints on the size of an object in sequences with motion towards or away from the camera. Efficient implementations on graphics processing units allow for interactive applications.

Furthermore, the thesis presents edge-based regularization for stereo reconstruction. It will be shown how image edge information can be incorporated in convex relaxation methods for stereo reconstruction for finding globally optimal solutions. In addition, a convex optimization method for multi-view stereo reconstruction with octrees for high-resolution 3D models is presented. Continuous formulations allow for memory-efficient implementations and avoid metrication errors. Experiments show that reconstruction results can be substantially improved when image edges are considered. The methods are applied to image-based plant phenotyping of grapevine and barley. Applications shown include computations of volumetric information for fruit-to-leaf ratios and monitoring of grapevine growth.

Acknowledgements

I express my gratitude to Prof. Dr. Daniel Cremers for supervising my thesis, and introducing me to the field of computer vision. I further thank him for his guidance which was very valuable for the thesis. Moreover, I thank Prof. Dr. Dr. Wolfgang Förstner for being the second referee and for the helpful suggestions that improved this thesis. I also thank him for the fruitful discussions and cooperation in the context of the CROP.SENSe.net project. I further thank Prof. Dr. Bjoern Menze for refereeing the thesis defense and Prof. Dr. Nils Thuerey for being the committee chairman.

Furthermore, I would like to thank the colleagues and co-authors whom I worked with during the last years. Special thanks goes to my collaborators Dr. Katja Herzog, Dr. Jürgen Sturm, Dr. Kalin Kolev and Frank Steinbrücker for the various interesting, fruitful and pleasant cooperations. Further thanks goes to the members of the TUM Computer Vision Group for the friendly working atmosphere. I also thank Dr. Henrik Schumann for fruitful discussions and for providing the image data for the experiments presented in Sec. 7.5, and Christian Kerl for capturing the quadcopter image data of Fig. 5.2.

Parts of this work were supported by the AgroClustEr: CROP.SENSe.net (FKZ 0315534D) which is funded by the German Federal Ministry of Education and Research (BMBF). I am grateful for this funding.

A special thanks goes to my friends and family, especially Horst and Felix.

Contents

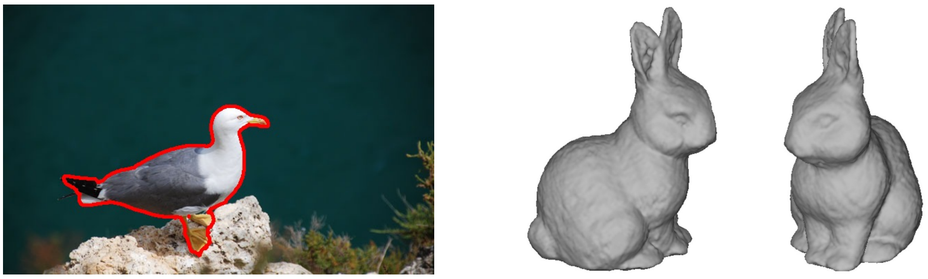
Chapter 1

Introduction

Research in the field of computer vision aims at enabling computers to understand visual impressions. At its core a digital image is merely a matrix of intensity values representing the amount of incident light at the moment of recording. Methods developed in computer vision interpret these images and draw conclusions about the depicted scenes. These conclusions include recognition and detection of known objects, segmentation of objects (from each other) and understanding the 3D structure of the scene. In the case of image series or videos additional information about the speed or change over time of objects may be gathered.

The importance of computer vision is increasing in numerous diverse areas due to the growing automation of industrial and everyday life. About a decade ago, computer vision methods were mainly dominant in industry and research. Stationary robots in factories were able to automatically fulfil specific tasks in a relatively static environment. Due to substantial advances both in algorithms and the computational power of smaller and smaller computers, computer vision has spread to diverse new fields of applications. During the recent years computer vision has entered the every day life. Today, smartphones are able to recognize objects and people and can automatically read text from photographs. Recently developed driver assistance systems are able to detect obstacles, traffic signs, and pedestrians and warn the driver, accordingly. Optical park distance control are becoming a standard in newly manufactured cars. The first autonomously driving cars are tested in public space and depend heavily on computer vision methods.

Since computer vision methods have entered the area of video game consoles, a new generation of low-cost consumer depth cameras is available, which in return are also widely used in computer vision research. RGB-D cameras like the Kinect capture colour images with additional depth information from an infra-red sensor. Image-based video game control is used to



(a) 2D image segmentation (b) 3D surface reconstruction

Figure 1.1: Examples for shape optimization: A shape can be represented by (a) a contour in the case of 2D image segmentation or (b) a surface in the case of 3D stereo reconstruction.

detect persons in camera images and track gestures and motion.

Computer vision is a fast-growing area of research. Novel applications include augmented and virtual reality. The field of autonomously navigating robots is increasing due to the advances of robust computer vision algorithms for localization and 3D scene understanding. Both fields may change the world we know today drastically.

A fundamental concept, which is one of the basics of the applications mentioned above, is shape optimization. The idea is to find optimal shapes describing the contours of objects in the 2D images or the 3D surface of objects in the real world. This thesis will show advances in the area of shape optimization, while the focus is on convex optimization methods for different types of image segmentation and stereo reconstruction.

1.1 Shape Optimization

Shape optimization is the problem of finding a shape which is optimal with regard to a certain cost function. It is a fundamental problem in computer vision and has applications to image segmentation and dense 3D reconstruction. In the context of image segmentation the problem of shape optimization can correspond to finding a contour surrounding the objects depicted in an image. In the context of 3D reconstruction it can correspond to the surface of the scene to reconstruct. Fig. 1.1 shows two examples where the optimal shape is represented by a contour in an image (Fig. 1.1 a) and a reconstructed 3D surface (Fig. 1.1 b).

1.1.1 Image Segmentation

Image segmentation is the meaningful segmentation of an image domain into pairwise disjoint regions. Ideally the regions fulfil a certain inner-regional similarity and inter-regional dissimilarity, which can for example be based on intensity or color values in the scene, or on texture, distance or motion. Usually it is desired that the regions correspond to the objects depicted in the scene. An additional constraint often used is that the contours of the regions are aligned with the edges of the image because these often correspond to the object contours. The task of image segmentation is usually an *ill-posed* problem which means that the solution is not unique, and is not always trivial, even for the human eye.

Image segmentation is a general concept that can be found in numerous applications of diverse areas. One of the most common applications is medical image analysis, where segmentation methods can be used amongst others to segment organs or identify anomalies in MRI or CT images. Other applications include the analysis of aerial images to distinguish vegetation from urban areas and optical character recognition for text scanning.

Since image segmentation is a highly non-unique task, shape priors are an established way to impose prior knowledge about the objects depicted in the scene. Shape constraints can help to scale down the set of possible solutions and help to reconstruct the desired shape. For example in the case of segmentation of buildings a constraint on rectangular shapes can be imposed. The most common assumption is the smoothness prior that demands that contours should be short in order to prefer compact objects and avoid overfitting due to noise or color ambiguities. Applications of shape constrained image segmentation include detection of known objects and object tracking in image sequences. Object tracking is the problem of following the contour of an object in a sequence of images. Usually the contour is given in the first frame of the image sequence, e.g. from a segmentation method or user input. The task of object tracking is then to follow the object in the subsequent frames. Object tracking has applications in monitoring of moving objects in video sequences including people tracking, car tracking, optimization of camera control in sports games i.e. in order to follow the ball. In the case of a moving camera, self-localization is an additional task.

1.1.2 Stereo Reconstruction

The goal of stereo reconstruction is the inference of the 3D scene structure from a set of 2D images depicting the scene from different view points. At least two images are needed for a stereo reconstruction, more images are used for multi-view stereo reconstruction. The aim of dense reconstruction is the reconstruction of a dense surface of the object to reconstruct. Dense reconstruction has the advantage that it yields not only 3D locations but additional volumetric and surface information. A dense surface can be computed from a complete segmentation of a discretized volume into object voxels and empty background voxels, whose contours form the surface of the object to reconstruct.

The problem of inferring 3D structure from 2D projections is an *ill-posed* problem. Due to the perspective projection in the camera capturing process, one dimension is lost and re-projection is not unique.

A commonly used representation for dense stereo reconstruction is a depth map. A depth map assigns to each pixel in an image the depth of the depicted object point, i.e. the distance to the camera position. For depth map reconstruction at least two images are needed with overlapping viewing range. For a full 3D reconstruction on the other hand, each object point must be visible in at least two images.

Depth map reconstructions are essential for robotic applications. They are used for mobile robots for localization and mapping and for stationary robots in factory automation processes. Other applications include visualizations in 3D movies and driver assistance where they can help to estimate distances to obstacles on the street. Newer applications include augmented reality applications and camera-based gesture recognition for human-machine interaction in video game consoles.

1.2 Related Work

Image segmentation and stereo reconstruction are well-studied problems in computer vision. Variational methods are widely used to compute solutions to these problems. They allow for one-step solutions with less free parameters compared to multi-step methods. Continuous and discrete optimization methods are two counterparts in variational optimization. For continuous representations, convex relaxation methods are an established method for finding global optima of continuous optimization problems. Advantages and disadvantages of the methods will be briefly reviewed in this section.

1.2.1 Variational Methods for Shape Optimization

Variational methods aim at minimizing or maximizing real-valued functionals. They provide a powerful tool for well-defined mathematical representations of image analysis problems, and have become an established formalism to compute single-step solutions to various computer vision problems including shape optimization problems such as image segmentation and 3D reconstruction.

The first variational methods for image segmentation were presented in the late 1980's with the edge-based *Snakes* functional in 1988 [77] and the region-based *Mumford-Shah* functional in 1989 [107]. The *Snakes* model, also called the *active contours* model, is an energy functional defined on the contour of the shape which is optimized locally to draw an initial contour to the image edges. The *Mumford-Shah* energy is defined on the regions and assumes inner-regional similarity and inter-regional dissimilarity of the intensity values of an image. The two different approaches have become merged together, as various methods nowadays incorporate both edge-based and regional terms [29].

While the *Snakes* and *Mumford-Shah* functionals were originally defined for explicit contour representations, implicit contours are the dominant method today. Explicit contours are based on a parametrization with a fixed number of control points that move through the image domain during optimization. Thus, the contour has a fixed resolution and topology, furthermore frequent regridding steps are necessary during the optimization to obtain uniform spacings between the contour points. Implicit contours on the other hand are based on an indicator or level set function that is defined on the whole image domain. The contour is not stored explicitly but is represented by a level set of the function. Implicit contour representations for image segmentation were presented with the *Geodesic active contours* [30] in the middle of the 1990s. They allow for topological changes of contours and efficient implementations. Furthermore implicit representations enable global optimization of certain functionals. Respective optimization methods include discrete optimization for example with graph cuts [26, 67] and continuous optimization with level sets [36, 113] and convex relaxations [31, 34].

1.2.2 Continuous and Discrete Representations

Continuous and discrete representations are two counterparts in variational functional optimization. Digital images are a discrete representation of a continuous scene. Graph cut methods are an established approach where the optimization functional is formulated on a discrete domain. Efficient algorithms for graph cut segmentations enable global optimization in polynomial computation time [26]. However, it was shown that discrete formulations like graph cut methods can yield contours that show a bias towards the orientations of the neighborhood connectivity of the underlying grid [5]. Continuous representations can help to avoid these metrication errors. Early methods using a continuous representation include level set formulations while more recent methods are based on convex relaxation of indicator functions. Level sets were introduced to computer vision in the work of Osher and Sethian in 1988 [113]. Region-based image segmentation with level sets was presented by Chan and Vese in 2001 [36]. Level set formulations have the advantage that the continuous formulation and consistent discretizations can enable arbitrarily accurate resolutions without discretization artefacts. However, optimization is performed only locally. More recently, convex relaxation methods based on the total variation norm have become popular since they enable continuous global optimization. Convex relaxation for image segmentation has been presented in 2005 by Chan et. al. [34] and Chambolle [31].

1.2.3 Convex Relaxation Methods

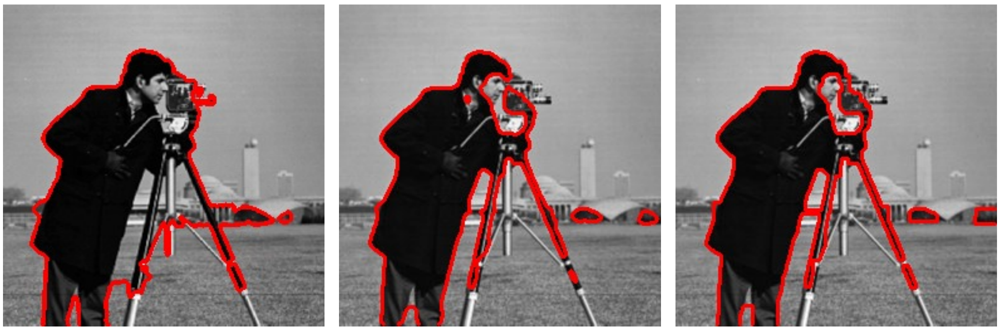
Convexity is a favorable property of energy functionals, since the high-dimensional functionals are usually optimized with local optimization techniques. If the functional to be minimized is convex, a global optimum can be found with local optimization, independent on initialization. For non-convex functionals either a good initial estimation is necessary or the optimization gets stuck in a local optimum.

Convex relaxation techniques have become a popular approach to a variety of image segmentation problems as they allow to compute solutions independent of initializations. Hence they are a step towards unsupervised methods that can operate on fully automated systems.

The total variation (TV) norm was introduced to computer vision by Rudin, Osher and Fatemi in 1992 in the context of image denoising [124]. A binary image segmentation method based on the total variation norm was published almost simultaneously in 2006 by Chan, Esedoglu and Nikolova [34] and Chambolle [31]. The methods enable continuous globally optimization.

Since then, the total variation norm has become a dominant factor for continuous variational methods in computer vision. The main reason is its convexity that allows for global optimization in contrast to the local optimization of level sets. Various computer vision problems can be solved globally optimal using convex relaxation methods based on the total variation norm – including image segmentation [34], disparity map reconstruction [120] and multi-view 3D reconstruction [10].

Fig. 1.2 shows a comparison for image segmentations with level sets (continuous local optimization), total variation (continuous global optimization) and graph cuts (discrete global optimization). The level set segmentation (Fig. 1.2 (a)) is dependent on the initialization, hence the optimization can get stuck in local optima. Starting from an initial contour, the contour is



(a) Level Set (b) Total Variation (c) Graph Cut

Figure 1.2: Image segmentation with different types of implicit contour representations. (a) Continuous optimization with level sets can get stuck in local optima. (b) Total variation minimization enables continuous and global optimization due to convexity. (c) Discrete optimization with graph cuts yields global optima but contours prefer the orientations of the underlying grid.

optimized until a local optimum is reached. In the case of Fig. 1.2 (a) this results in a contour that stops the optimization at image regions that are the best match in a local neighborhood. Total variation minimization (Fig. 1.2 (b)) on the other hand is independent on the initialization. Due to the convexity of the functional, a global optimum can be found with local optimization methods. Graph cut methods (Fig. 1.2 (c)) are able to compute global optima without initialization. The discrete formulation can result in metrication errors caused by the bias to the underlying grid, resulting in a preference to contours that are parallel to the image axes. A more detailed comparison of discrete and continuous optimization methods can also be found in Chapter 3.

1.2.4 Image Segmentation

The convex formulation of the two-region image segmentation problem [31, 34] allows for continuous global optimization of binary segmentation problems. For multi-label segmentation the problem is no longer convex. Total variation based methods for multi-label image segmentation have been presented in [119], [93] and [148] that approximate globally optimal solutions. An overview of different optimization methods for multi-label image segmentation has been presented in [112].

Shape priors are an established way to stabilize segmentation results. A global optimization method for image segmentation with shape priors has been presented in [128]. Most methods are based on a previous learning of reference shapes [68, 39, 52] and try to find the silhouette of the shape in a new image. More general formulations include shape constraints based on the low-level features of a shape. Graph cut based formulations have been presented for segmentation of compact objects [41] and star-shaped objects [143]. A shape prior for convexity was presented in [63], however no guarantee can be made that the results are global optimal solutions. A convex formulation for connectivity constraints has been presented in [135].

1.2.5 Stereo Reconstruction

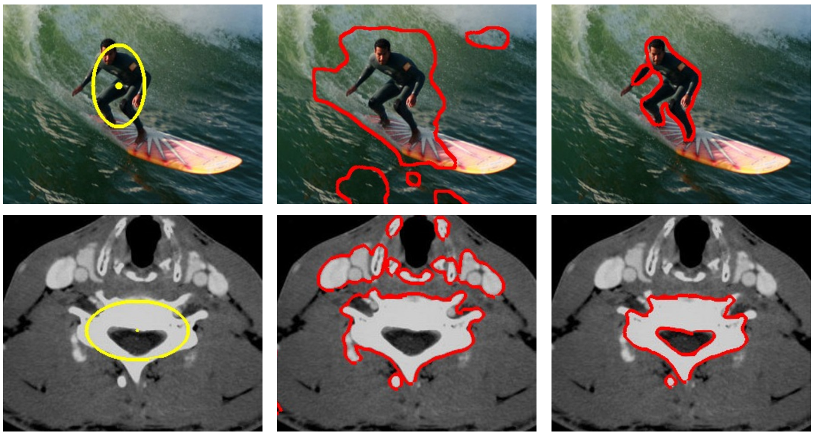
Efficient algorithms for stereo reconstruction of depth maps include the semi-global matching method presented in [71]. Global optimization for stereo reconstruction was first proposed for discrete optimization with Markov Random Fields. In [72] it was shown that certain multi-label problems can be solved globally optimal, if the labels can be ordered. This applies to stereo reconstruction, because the labels can be ordered by depth. A convex relaxation method based on this work was presented in [120]. The multi-view stereo reconstruction problem was formulated with convex relaxation in [9]. An extension that integrates surface normals based on anisotropy has been presented in [83].

1.3 Contributions

This thesis is about the study of convex relaxation methods for shape optimization, focused on image segmentation and stereo reconstruction. The main contributions are novel methods for image segmentation with moment constraints, scale-aware object tracking, edge-based stereo reconstruction and high-resolution volumetric multi-view reconstruction. The convex formulations allow for continuous global optimization, independent of initializations. Applications shown in this thesis include interactive segmentation, medical image analysis, object tracking in RGB and RGB-D sequences and image-based plant phenotyping for grapevine and barley. In the following, a short overview of the contributions is given.

1.3.1 Experimental Comparison of Continuous and Discrete Shape Optimization Methods

Continuous and discrete optimization are two diverse directions in optimization methods used in computer vision. Both representations are popular and widely used for shape optimization, including image segmentation, stereo



(a) Input Ellipse

(b) Segmentation with color only

(c) with moment constraints

Figure 1.3: Image segmentation with area, centroid and covariance constraints, for interactive segmentation (first row) and medical imaging (second row). The shape constraints for the segmentation are derived from the input ellipse as an intuitive user interface. The convex formulation allows for globally optimal solutions.

reconstruction and multi-view reconstruction. In Chapter 3 a quantitative experimental comparison of continuous optimization based on convex relaxation and discrete optimization with graph cuts is presented. The methods are compared with respect to computation times, memory consumption and accuracy, using the example of image segmentation and multi-view 3D reconstruction. This allows for an objective discussion of the strengths and limitations of both approaches.

1.3.2 Convex Moment Constraints for Image Segmentation and Object Tracking

Shape constraints for image segmentation are usually learned from a set of reference shapes which makes a previous learning step necessary. More general shape constraints based on the low-level features of a shape like compactness, convexity or connectivity constraints avoid this preprocessing step. In Chapter 4, a convex formulation for image segmentation with moment constraints is presented. The proposed method allows to constrain the approximate dimensions of a segmentation like size (area constraints), location (centroid constraints) and relation of width to height (covariance constraints) (see Fig. 1.3). Experiments show that the convex moment constraints can stabilize results for interactive image segmentation, medical image analysis and object tracking.

1.3.3 Scale-Aware Object Tracking in RGB-D Sequences

Constraining the area of a shape during object tracking in an image sequence yields robust results for parallel motions. However, if the object moves towards or away from the camera, the *projected* size of the object in the image plane changes, although its *absolute* size remains constant. In Chapter 5, a scale-aware method for object tracking with convex 3D shape constraints is presented, that allows to robustly track objects in RGB-D image sequences with motion towards or away from the camera.

1.3.4 Edge-Based Regularization for Stereo Reconstruction

Stereo reconstruction for depth map estimation is usually based on regional information measuring point-wise matching costs, while image edges are usually neglected. Image edges can be computed from the image gradient and can contain valuable information as they are often aligned with the boundaries of the objects depicted in the scene. In Chapter 6, a method for incorporating image edge information to stereo reconstruction based on anisotropic regularization is presented. The convex formulation allows for accurate depth map reconstructions compared to standard regularization.

1.3.5 Stereo Reconstruction for Plant Phenotyping

Since plant researchers are increasingly working on large numbers of samples, manual measurements are not sufficient anymore. Image analysis has become a popular method for phenotyping as it provides an efficient tool for automated plant measurements. However, as many methods are not completely automated or not robust enough to be used on outdoor images captured directly in the field, the so-called *phenotypic bottleneck* still prevents high-throughput analysis on large data bases. In Chapter 6, a robust method for phenotyping of grapevine is presented. The method uses convex formulations for depth reconstruction and color based segmentation that enable estimation of 3D leaf areas. The method was successfully applied to monitoring of grapevine growth and computations of fruit-to-leaf ratios. Robustness is shown for images taken from a moving platform directly in the field.

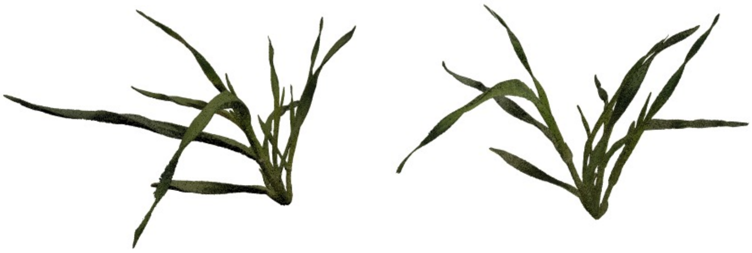


Figure 1.4: Volumetric 3D reconstruction of a barley, computed from RGB images. The dense surface is optimized in the leaf nodes of deepest level in an octree.

1.3.6 High-Resolution Volumetric Multi-View Reconstruction with Octrees

Sparse methods for 3D reconstruction enable representation of high-resolution point clouds however provide no volumetric or surface information. Dense methods on the other hand are usually defined on a uniformly discretized voxel grid, limiting resolutions drastically. In Chapter 7, a volumetric approach for high-resolution dense surface reconstruction from images is presented. High resolutions are achieved by using the octree data structure. Based on a segmentation in the visual hull of the object, the optimization problem can be formulated convex. Fig. 1.4 shows an example for a high-resolution volumetric 3D reconstruction.

1.4 Outline

The remainder of the thesis is organized as follows:

Chapter 2 gives an overview of the relevant background, definitions and fundamentals.

Chapter 3 gives an overview of related work on continuous and discrete shape optimization methods. Furthermore it shows an experimental comparison of the methods with respect to run-time, memory consumption and accuracy.

Chapter 4 presents image segmentation with moment constraints. It shows how segmentation results can be improved by imposing shape constraints on the lower order moments of a shape, integrated in a convex formulation for variational shape optimization. Applications shown include interactive image segmentation and medical image analysis.

Furthermore, an extension of the method to object tracking in image sequences is presented.

Chapter 5 shows an extension of the moment constraints tracking presented in Chapter 4 for scale-aware object tracking in 3D using RGB-D image sequences. The chapter shows a robust method for object tracking based on convex shape constraints.

Chapter 6 describes convex relaxation methods for stereo reconstruction for disparity map estimation. An extension to anisotropic regularization is shown for improving reconstruction accuray especially at the object boundaries. Furthermore, an application to phenotyping of grapevine growth is presented where the method was successfully employed on real world images.

Chapter 7 describes convex relaxation for multi-view stereo reconstruction as well as a memory-efficient implementation in octrees allowing for high-resolution volumetric 3D reconstructions.

Chapter 8 concludes the thesis with a summary of the main results and contributions as well as an outlook to possible future work.

Chapter 2

Background

Variational methods provide well-defined mathematical formulations for image analysis. They are a powerful and versatile tool which have become established in computer vision research. This chapter presents the basics and early work on variational image segmentation in Sec. 2.1, Sec. 2.2 and Sec. 2.3, as well as an introduction to total variation in Sec. 2.4, which provides a key concept for convex relaxation methods. The presentation of early works in Sec. 2.3 will focus on two functionals: the *Mumford-Shah* functional which is the basis of many segmentation methods today, and the *Chan-Vese* functional which introduced a level set formulation of the Mumford-Shah functional leading to the now widely used implicit contour representations in shape optimization. Sec. 2.5 describes convexity as a favorable property in energy minimization.

2.1 Introduction

Images and segmentations of images can be considered as continuous functions. This section describes the representations for images, segmentations and shapes that are used in this thesis.

2.1.1 Continuous Image Representation

A digital image is a matrix of vector-valued intensity values, called pixels. In this thesis, two dimensional images will be considered. The dimension b of a pixel corresponds to the number of channels in the image. For example, an intensity image has $b = 1$ channel, a color image in RGB space has $b = 3$ channels, this corresponds to one channel each for red, green and blue. RGB-D images have an additional depth channel with $b = 4$.

The concept of pixels can be transfered to 3D space: Here a volume can be discretized into a uniform grid of voxels. This is the representation used for 3D volumes in this thesis.

Although digital images are discrete, they can be represented in a continuous formulation. An image I can be interpreted as a continuous function I defined on a continuous domain $\Omega \subseteq \mathbb{R}^d$:

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

where both the spatial image domain Ω and the range of intensity values $\mathbb{R}^b$ are represented in a continuous formulation. The continuous formulation allows for mathematical analysis with well-studied numerical optimization methods for continuous functions.

2.1.2 Image Segmentation

Image segmentation is the partitioning of the image domain into meaningful regions. Hence, the segmentation of $\Omega \subset \mathbb{R}^2$ into a set of n pairwise disjoint regions Ω_i can be defined as

$$\Omega = \bigcup_{i=1}^{n} \Omega_i, \quad \Omega_i \cap \Omega_j = \emptyset \;\; \forall i \neq j. \tag{2.2}$$

Usually the regions should have a certain inner-regional similarity and inter-regional dissimilarity, e.g. with respect to their intensity or color distributions. Various other constraints on the segmentation can be defined, for example on the shape or size of the regions.

2.1.3 Shape Representation

A shape can be represented in 2D by a contour in an image, or by a surface in 3D. The contour C of a segmentation is defined as the border of the segmentation: $C = \bigcup_{i=1}^{n} C_i$ where $C_i = \partial\Omega_i$ is the contour of region Ω_i. Contour representations can be parametrized explicitly or represented implicitly with a corresponding indicator or level set function.

A parameterized contour can be represented e.g., by splines and stored explicitly by a set of n control points $x_1, ...x_n$. This representation is suitable for fixed contours, however if the contour is optimized, regridding steps can be necessary. An early work on image segmentation using an explicit contour representation for variational image segmentation is the *Snakes* method [77].

For an implicit representation, the contour C_i of a region Ω_i is represented with a higher dimensional auxiliary function. In the case of a level set representation this function is called a level set function which has positive values inside the region it represents and negative values outside. The contour C_i is then defined as the 0-level-set. A more detailed overview of level set methods is given in Sec. 2.3.2. A binary indicator function $u_i : \Omega \to \{0, 1\}$ representing the region is given by

$$u_i(x) = \begin{cases} 1 & \text{if } x \in \Omega_i \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The contour is defined as the transitions between 0 and 1. An implicit contour representation allows for discretization with a fixed resolution due to the underlying pixel grid, which avoids control point regridding. Implicit contours are used in level set formulations [36] and convex relaxations [35].

In this thesis, implicit contour representations will be considered due to their advantages over explicit contours with respect to optimization methods.

2.1.4 Image Derivatives and Discretizations

The gradient of an image measures the change of intensity values in an image. It has typically high absolute values at image edges and corners and low absolute values in relatively homogenous regions. Image edges and corners contain important information about the scene depicted in the image, as they typically coincide with the boundaries of the objects depicted in the scene. The gradient ∇u of a function $u : \mathbb{R}^d \to \mathbb{R}$ is defined as the vector

$$\nabla u := \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}, \tag{2.4}$$

where $\partial u / \partial x_i$ is the partial derivative of u with respect to x_i.

The gradient norm is the length of the gradient vector:

$$|\nabla u| = \sqrt{\sum_{i=1}^{d} \left(\frac{\partial u}{\partial x_i} \right)^2}. \tag{2.5}$$

The divergence div is an operator that maps a vector $p \in \mathbb{R}^n$ to a scalar:

$$\operatorname{div}(p) := \sum_{i=1}^{d} \frac{\partial p_i}{\partial x_i}. \tag{2.6}$$

The Laplace operator Δ is defined as the divergence of the gradient and is given by the sum of second derivatives of the components of u:

$$\Delta u := \operatorname{div}(\nabla u) = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}. \tag{2.7}$$

When image derivatives are discretized, it is important that the discretizations are consistent, especially when multiple derivatives are concatenated. Otherwise discretization artefacts can appear.

Discretization of the partial derivatives $\frac{\partial u}{\partial x_i}$ can be formulated using forward differences D^+, backward differences D^- or symmetric differences D^0. For a two-dimensional domain Ω the discretizations in x_1 direction are defined as

$$\begin{aligned} D^+ u_{x_1} &= \tfrac{1}{h}\left(u(x_1+h, x_2) - u(x_1, x_2)\right), && (2.8)\\ D^- u_{x_1} &= \tfrac{1}{h}\left(u(x_1, x_2) - u(x_1-h, x_2)\right), && (2.9)\\ D^0 u_{x_1} &= \tfrac{1}{2h}\left(u(x_1+h, x_2) - u(x_1-h, x_2)\right), && (2.10) \end{aligned}$$

where h is the width of a pixel. In a discrete implementation, it is usually $h = 1$. For $h \to 0$ the forward difference converges to the continuous definition of derivatives. Discretizations in x_2 direction are analogously defined.

Discretization of the Laplace operator Δ (2.7) can be either achieved with the second derivatives directly or by a concatenation of two first derivatives. Using the latter version, it is important to use a consistent discretization of the derivatives. A common consistent choice is the forward difference for the gradient and backward difference for divergence:

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} \approx D^-\left(D^+ u_{x_1}\right) + D^-\left(D^+ u_{x_2}\right). \qquad (2.11)$$

2.1.5 Diffusion

The diffusion equation describes a process that distributes intensities of the function u spatially:

$$\partial_t u = \operatorname{div}(g \nabla u), \qquad (2.12)$$

where t is a time step and g describes the speed of diffusion. $g \in \mathbb{R}$ is called *diffusivity*. For $g = 1$ the right hand side of (2.12) corresponds to the Laplace operator (2.7). Diffusion of an image yields a smoothed version of the image, while the number of iterations t corresponds to the grade of smoothing. For $t \to \infty$ the diffusion process converges to the average intensity value of the image.

The diffusion process is called *linear* if g is a constant scalar value for all points. If g depends on location x, the process is called *inhomogenous* diffusion. If g depends on u, it is called *non-linear* diffusion. If g is a matrix it is called *anisotropic* diffusion, and g is called *diffusion tensor*.

2.2 Energy Minimization

In this thesis functionals of the following form are considered

$$E(u) = E_{\text{Data}}(u) + \lambda E_{\text{Smoothness}}(u), \tag{2.13}$$

where E is an energy functional consisting of a data term $E_{\text{Data}}(u)$ and a regularizer term $E_{\text{Smoothness}}$, weighted with a smoothness parameter $\lambda \in \mathbb{R}$. The minimum of E is a function $u := (u_1, \dots, u_n)$ that fulfils a data fidelity implemented in the data term D and smoothness prior implemented in the regularizing term R. In the following, the data and smoothness terms are presented in more detail.

2.2.1 Data Term

The data term measures similarity to the input image by a data fidelity model.

Data terms can be based on mean intensities or mean colors $c_i \in \mathbb{R}^b$ where b is the number of color channels, for example $b = 1$ for intensity images and $b = 3$ for RGB images:

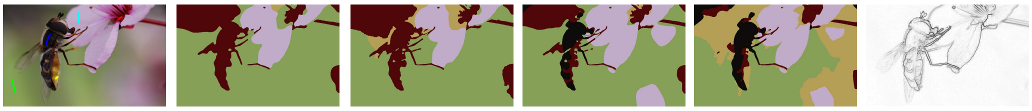
$$E_{\text{Data}}(u) = \sum_{i=1}^{n} \int_{\Omega} (I(x) - c_i)^2 u_i \, \mathrm{d}x. \tag{2.14}$$

Another way to compute color-based data terms is to use the color distributions inside a region with histograms. A histogram $p_i : \mathbb{R}^b \to [0, 1]$ encodes the distribution of intensity or color values while the range of intensities is discretized to a fixed number of bins. Data terms can be computed using b-dimensional histograms p_i in the following way:

$$E_{\text{Data}}(u) = \sum_{i=1}^{n} \int_{\Omega} -\log(p_i(I(x)))u_i \, \mathrm{d}x. \tag{2.15}$$

Fig. 2.1 shows a comparison of different methods for the data term for an example image segmentation. The data terms for the regions are computed from user seeds, i.e. pixels that were manually labelled by a user. In the depicted example only the data term based on RGB histograms is able to segment the bee from the background. The input image is from the *IcgBenchmark* data set [125].

More advanced data terms include the incorporation of co-occurrence statistics [89] or label costs [142]. In [110] the authors consider the location of pixels marked by a user to compute higher dimensional histograms. This allows for spatially dependent data terms, in addition to color distribution based terms.



(a) Image with seeds (b) Intensity means (c) Intensity histograms (d) RGB means (e) RGB histograms (f) Edge weights

Figure 2.1: Segmentation from user seeds (a) for four different data terms (b-e) and an edge weighting function (f).

2.2.2 Regularization Term

The regularization term $E_{\text{Smoothness}}(u)$ measures the smoothness of the contour. Usually it is demanded that the contour should be smooth and short. The balance of the data term and regularization term can be varied by choosing the smoothness parameter λ accordingly.

Minimizing the contour length is a common choice for the regularization term. Often it an additional requirement demands that the contour should be aligned with the object boundaries. Image gradients indicate edges of the depicted objects. This can be used to weight the contour length with a function that has low values at regions where the image gradient norm $|\nabla I|$ is high, and high values where it is low. The edge detection function $g : \Omega \to [0, 1]$ should be monotonically decreasing to yield high values for small image gradients, which indicate homogeneous regions in the image, and small values for high image gradients which indicate object boundaries. The following functions g_1 and g_2 are commonly used edge detectors:

$$g_1(x) = \frac{1}{1 + \beta |\nabla I_\sigma(x)|^2} \tag{2.16}$$

with the parameters $\beta \in \mathbb{R}$ and $\sigma \in \mathbb{R}$. I_σ is a Gaussian smoothed version of the input image and σ is the standard deviation of the Gaussian.

$$g_2(x) = \exp(-\gamma |\nabla I_\sigma(x)|^\alpha) \tag{2.17}$$

with parameters $\gamma, \alpha, \sigma \in \mathbb{R}$. An example of the edge weight function g_2 is shown in Fig. 2.1 (f). The parameters were set to $\gamma = 5$, $\alpha = 1$ and $\sigma = 3$.

The regularization term can also be weighted with a tensor, yielding an anisotropic regularization, as for example used in [145] for optical flow or in [127] for image compression.

2.3 Variational Methods for Image Segmentation

The establishment of variational methods for image segmentation started in 1988 with the Snakes functional [77] and 1989 with the Mumford-Shah functional [107]. The Mumford-Shah functional is the basis for many image segmentation methods today. Implicit contour representations were introduced to image segmentation with the level set method [113, 36]. In the following, the Mumford-Shah functional and its formulation with level sets by Chan and Vese [36] will be discussed in more detail.

2.3.1 Region-based Segmentation by Mumford and Shah

There exist different versions of the Mumford-Shah functional. The general form describes a piecewise smooth approximation of an input image while a specialization is the piecewise constant approximation.

The piecewise smooth Mumford-Shah functional was presented in 1989 [107] as

$$E(u,C) = \int_{\Omega} (I-u)^2 \,\mathrm{d}x + \lambda \int_{\Omega \setminus C} |\nabla u|^2 \,\mathrm{d}x + \nu \,|C| \tag{2.18}$$

The energy depends on a piecewise smooth approximation u of the input image I and a contour C. The minimum of (2.18) is a piecewise smooth approximation $u : \Omega \to [0,1]$ of an input image I. Since functional (2.18) contains both u and its derivative ∇u, it is a partial differential equation whose minimum cannot be computed explicitly. Numerical methods are needed to solve it, and furthermore convex formulations to obtain a global minimizer.

A special case of the Mumford-Shah functional is the piecewise constant approximation with n regions $\Omega_1, \ldots, \Omega_n$ minimizing

$$E(u,C) = \sum_{i=1}^{n} \int_{\Omega_i} (I(x) - u_i)^2 \,\mathrm{d}x + \nu \,|C| \tag{2.19}$$

where u_i is constant for all $x \in \Omega_i$.

2.3.2 Implicit Contour Representation with Level Sets

Level sets are an important prior work for convex relaxation methods. They introduced the concept of an implicit contour representation by the use of a higher dimensional variable ψ into variational segmentation methods. Level sets were originally introduced to computer vision and computer graphics by Osher and Sethian in 1988 [113], and revived 1998 with a level set formulation of the two-label piecewise constant Mumford-Shah functional by Chan and Vese [36].

The Chan-Vese functional [36] is a level set formulation of the piecewise constant Mumford-Shah functional (2.19) with two regions that correspond to foreground and background, respectively. It uses an implicit contour representation with a level set function $\psi : \Omega \to \mathbb{R}$:

$$\begin{aligned} E(c_1, c_2, \psi) = &\int_\Omega (I(x) - c_1)^2 H(\psi) + (I(x) - c_2)^2 (1 - H(\psi)) \, \mathrm{d}x \\ &+ \nu \int_\Omega |\nabla H(\psi(x))| \, \mathrm{d}x + \mu \int_\Omega H(\psi(x)) \, \mathrm{d}x. \end{aligned} \tag{2.20}$$

The level set function ψ has positive values in the foreground region and negative values in the background. $H : \Omega \to \{0, 1\}$ is called *Heaviside* function, and indicates for each pixel x if it belongs to foreground ($H(\psi(x)) = 1$) or background ($H(\psi(x)) = 0$). In practice, a smoothed version H_ϵ of H can be used to avoid the non-differentiability of H. The function E in (2.20) depends on two mean intensity values $c_1, c_2 \in \mathbb{R}$ of the foreground and background region, respectively.

The contour C is implicitly represented in (2.20) by the zero level set of ψ:

$$C = \{x \in \Omega \mid \psi(x) = 0\}. \tag{2.21}$$

The implicit representation also allows for topological changes of the contour.

2.4 Total Variation

The total variation norm (TV norm) plays an important role in continuous convex optimization. It is a widely used regularizer for shape optimization because of its convexity and edge-preserving property.

2.4.1 Total Variation Norm

The total variation norm (TV norm) of a function $u : \Omega \to \mathbb{R}$ with $\Omega \subseteq \mathbb{R}^n$ is defined as

$$TV(u) = \sup_{\varphi \in \Phi} \left\{ \int_\Omega u(x) \operatorname{div} \varphi(x) \, \mathrm{d}x \right\}, \tag{2.22}$$

$$\text{with} \quad \Phi := \left\{\varphi \in C^1(\Omega, \mathbb{R}) \,:\, |\varphi(x)| \leq 1, \forall x \in \Omega\right\} \tag{2.23}$$

For continuous differentiable functions u the TV norm is given by

$$TV(u) = \int_\Omega |\nabla u| \,\mathrm{d}x \tag{2.24}$$

A function u with $TV(u) < \infty$ is called a function with bounded variation. The space of functions with bounded variations is called BV.

2.4.2 Coarea Formula

The coarea formula [134] states that for functions u with bounded variation the TV norm equals the integration of the contour lengths of all level sets Γ_μ of u:

$$\int_\Omega |\nabla u| \,\mathrm{d}x = \int_\mathbb{R} \left(\int_{\Gamma_\mu} \mathrm{ds} \right) \mathrm{d}\mu = \int_\mathbb{R} \mathrm{Per}(\Omega_\mu) \,\mathrm{d}\mu \tag{2.25}$$

where $\mathrm{Per}(\Omega_\mu) = \int_{\Gamma_\mu} ds$ is the perimeter of the contour of the level set Γ_μ, $\Omega_\mu := \{x \in \Omega \,:\, u(x) > \mu\}$, and Γ_μ is the boundary of Ω_μ. The coarea formula is especially important for the thresholding theorem that will be described in Sec. 3.2.1.

2.4.3 Total Variation for Binary Functions

A special case occurs when u is a binary function. This case is interesting because shape optimization methods are often a problem of binary or piecewise constant labellings. In these cases the TV norm provides the following property: The TV norm for binary functions is a measure for the length of the contour of a region, in 3D space it is a measure for the surface area of a shape. A binary function u is the indicator function $\mathbf{1}_u$ of a set $\Omega_{\mathbf{1}_u}$ defined as

$$u(x) = \begin{cases} 1 & \text{if } x \in \Omega_{\mathbf{1}_u} \\ 0 & \text{if } x \notin \Omega_{\mathbf{1}_u}. \end{cases} \tag{2.26}$$

The TV norm of a binary function u is a measure for the contour length of the corresponding region $\Omega_{\mathbf{1}_u}$:

$$\int_\Omega |\nabla u| = \int_\Omega |\nabla \mathbf{1}_u| = \mathrm{Per}(\Omega_{\mathbf{1}_u}). \tag{2.27}$$

This allows to measure dimensions of contours or, in the case of three dimensional labelling functions, surface areas.

This property makes the total variation norm an appropriate candidate for the regularizing term, since minimizing the TV norm of u yield short contours $C = \partial u$, or small surface areas in 3D, respectively.

2.4.4 Weighted Total Variation

The weighted TV norm [29] includes an edge detection function $g : \Omega \to \mathbb{R}$ as a weight to the contour length:

$$TV(u) = \int_\Omega g(x)|\nabla u| \, \mathrm{d}x. \tag{2.28}$$

The function g can have the function as an edge-detector in the case of image segmentation or a photo-consistency measure in the case of 3D shape reconstruction.

2.5 Convexity

A set $\Omega \subseteq \mathbb{R}^d$ is called *convex*, if

$$\forall x, y \in \Omega : \forall t \in [0,1] : tx + (1-t)y \in \Omega. \tag{2.29}$$

A function $f : \Omega \to \mathbb{R}$ is called *convex*, if

$$\forall x, y \in \Omega : \forall t \in [0,1] : f(tx + (1-t)y) \leq tf(x) + (1-t)f(y). \tag{2.30}$$

Convexity is a favorable property in energy minimization. Minimization problems in computer vision are often based on partial differential equations, i.e. equations that contain unknown functions and their partial derivatives. Therefore, they are often not explicitly solvable and numerical methods are applied to find a solution. These numerical methods usually find a local optimum of the respective functional. In the case of a convex functional, every local optimum is a global optimum. Hence, convex functionals defined on convex sets can be minimized globally optimal and independent of initializations. An overview of different numerical methods for energy minimization is given in Sec. 3.2.2.

2.6 Conclusion

This chapter gave an overview of basic concepts and early works that lead to the development of convex relaxation methods which will be described in the subsequent chapter. The Mumford-Shah functional was introduced as an early work for variational image segmentation, and its level set formulation, the Chan-Vese functional using an implicit contour representation avoids the limitations of parametrized curves such as regridding. The total variation norm was described which is the key concept that enables convex optimization of certain energy functionals including the Mumford-Shah functional. The next chapter will show how it can be used to formulate globally optimal optimization methods using convex relaxation, which is the basic concept for this thesis.

Chapter 3

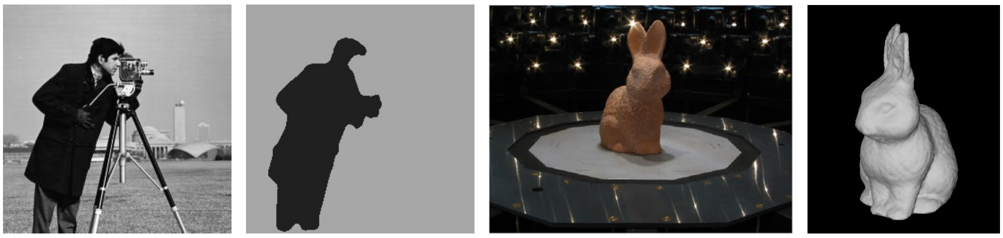
Continuous and Discrete Shape Optimization Methods

This chapter compares continuous and discrete shape optimization methods. In Sec. 3.1 a short introduction to shape optimization is given. Sec. 3.2 and Sec. 3.3 discuss the background and related work of continuous global optimization using convex relaxation and discrete shape optimization methods using graph cuts, respectively. In Sec. 3.4 a quantitative comparison of 2D and 3D reconstruction accuracies, memory requirements and computation times for 2D and 3D is presented. This allows for a discussion on strengths and limitations of both techniques. The result of this comparison determined the choice of optimization method for the methods presented in the subsequent chapters.

Parts of this chapter have been published in [5].

3.1 Introduction

Shape optimization occurs in numerous computer vision problems including image segmentation and multi-view stereo reconstruction. Following a series of seminal papers [77, 58, 20, 107, 49], functional minimization has become the established paradigm for these problems. This chapter is focussed on a certain class of labeling problems which can be globally optimized both in a spatially continuous framework using convex relaxation methods and in a spatially discrete setting. In the discrete setting the study of the corresponding binary labeling problems goes back to the spin-glas models introduced in



Input image

Intensity-based segmentation

One of 33 input images

Reconstructed object

Figure 3.1: Examples of shape optimization: Image segmentation and 3D reconstruction.

the 1920's [73]. This chapter is focussed on a class of functionals of the form:

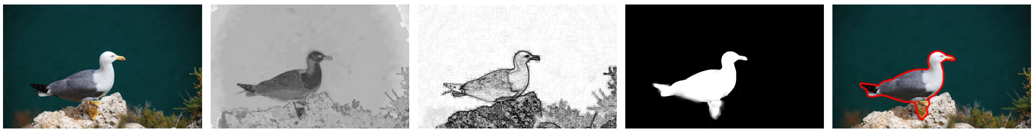
$$E(S) = \int\limits_{int(S)} f(x)\,\mathrm{d}x \;+\; \nu \int\limits_{S} g(x)\,\mathrm{d}S, \tag{3.1}$$

where S denotes a hypersurface in $\mathbb{R}^n$, i.e. a set of closed boundaries in the case of $2D$ image segmentation or a set of closed surfaces in the case of $3D$ segmentation and multi-view reconstruction. Here, $int(S)$ denotes the region enclosed by the hypersurface S. The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^+$ are application dependent. In a statistical framework for image segmentation, for example, $f(x) = \log p_{bg}(I(x)) - \log p_{ob}(I(x))$ may denote the log likelihood ratio for observing the intensity $I(x)$ at any given point x given that x is part of the background or the object, respectively.

The second term in (3.1) corresponds to an isotropic measure of area (for $n = 3$) or boundary length ($n = 2$), measured by the function g. In the context of image segmentation, g may be a measure of the local edge strength – as in the geodesic active contours [30, 79] – which energetically favors segmentation boundaries along strong intensity gradients. In the context of multi-view reconstruction, $g(x)$ is typically a measure of the inconsistency among different views of the voxel x, where low values of g indicate a strong agreement from different cameras on the observed patch intensity – see for example [49]. Fig. 3.1 shows shape optimizations using the examples of image segmentation and multi-view reconstruction.

Functionals of the form (3.1) can be globally optimized by reverting to implicit representations of the hypersurface S using an indicator function $u : \mathbb{R}^n \to \{0, 1\}$, where $u = 1$ and $u = 0$ denote the interior and exterior of S. The functional (3.1) defined on the space of surfaces S is therefore equivalent to the functional

$$E(u) = \int_{\mathbb{R}^n} f(x)\,u(x)\,\mathrm{d}x + \nu \int_{\mathbb{R}^n} g(x)\,|\nabla u(x)|\,\mathrm{d}x, \tag{3.2}$$



(a) Input image I

(b) Data term f

(c) Edge weight g

(d) Segmentation u

(e) Contour S

Figure 3.2: Image segmentation with convex relaxation. The input image I yields the regional data term f and edge weight function g. The segmentation u prefers regions where f is small and contours are aligned with g. The resulting contour S is the border of the segmentation u.

defined on the space of binary labelings u, where the second term in (3.2) is the weighted total variation norm which can be extended to non-differentiable functions in a weak sense. Note that the data term f can also take negative values, hence the trivial solution $u = 0$ is usually not a global minimizer of (3.2). Functional (3.2) is convex in u, even for non-convex data terms f and edge weights g, since f and g are independent of the optimization variable u. Hence, the first term in (3.2) is linear in u, and the second term is a weighted gradient norm, which is also convex. Due to the convexity of (3.2), a global optimum can be found with local optimization methods, using convex relaxation. The concept of convex relaxation will be described in Sec. 3.2.

Fig. 3.2 shows a visualization of the different parameters of functional (3.2). Here, the regional data term f is computed from the color values of input image I, and the edge weight function g is computed from the image gradient. Hence, the resulting segmentation u prefers small values of f and contours which correlate with the image edges. In Fig. 3.2 (b)–(d), dark regions indicate small function values and bright regions correspond to high values. The resulting contour S is the border of the segmentation u.

This experimental comparison is focussed on functionals of the type (3.1) since they allow for the efficient computation of globally optimal solutions of region-based functionals. There exist numerous alternative functionals for shape optimization, including ratio functionals [130, 74]. It was shown that some region-based ratio functionals can be optimized globally [85].

The functional (3.2) can be globally optimized in a spatially discrete setting: By mapping each labeling to a cut in a graph, the problem is reduced to computing the minimal cut. First suggested in [67], it was later rediscovered in [25] and has since become a popular framework for image segmentation [123] and multi-view reconstruction [144]. More recently it was shown in

[31, 34] that the same binary labeling problem (3.2) can be globally minimized in a spatially *continuous* setting as well. An alternative spatially continuous formulation of graph cuts was developed in [14].

This chapter presents a quantitative experimental comparison of spatially discrete and spatially continuous optimization methods for functionals of the form (3.2). In particular, the focus is on the quality and efficiency of shape optimization in discrete and continuous settings.

The outline of this chapter is as follows: In Sec. 3.2 continuous global optimization based on the total variation norm is presented. The thresholding theorem will be shown which states that convex relaxation methods can be used to obtain solutions to certain binary optimization problems. Furthermore, the section will present an overview of different numerical methods to solve the respective energy functionals. Sec. 3.3 will describe a discrete optimization method with graph cuts. Different metrics for approximating the length term will be discussed. These two concepts will be compared in Sec. 3.4 with regard to computation times, memory consumption and metrication errors using the example of image segmentation and 3D reconstruction. The section will show that continuous optimization based on the total variation norm can converge to minimal surfaces. Finally, Sec. 3.5 will conclude with a summary of the main results.

3.2 Continuous Optimization via Convex Relaxation

It was shown that the class of functionals (3.2) can also be minimized in a spatially continuous setting by reverting to convex relaxations [31, 34]. By relaxing the binary constraint and allowing the function u to take on values in the interval between 0 and 1, the optimization problem becomes minimizing the convex functional (3.2) over the convex set

$$u : \mathbb{R}^n \rightarrow [0, 1]. \tag{3.3}$$

Global minimizers u^* of this relaxed problem can be computed for example by gradient descent methods, or by more efficient numerical schemes such as multi-grid methods, coarse-to-fine strategies or linearization and fixed point iterations which will be detailed in Sec. 3.2.2.

3.2.1 Thresholding Theorem

The following theorem [134, 34] assures that thresholding the solution u^* of the relaxed problem provides a minimizer of the original binary labeling problem (3.2). In other words, the convex relaxation preserves global optimality for the original binary labeling problem.

Theorem 1. *Let $u^* : \mathbb{R}^n \to [0,1]$ be a global minimizer of the functional* (3.2)*. Then all upper level sets (i.e. thresholded versions)*

$$\Sigma_{\mu,u^*} = \{x \in \mathbb{R}^n \,|\, u^*(x) > \mu\}, \qquad \mu \in (0,1), \tag{3.4}$$

of u^* are minimizers of the original binary labeling problem (3.1)*.*

Proof. Using the layer cake representation of the function $u^* : \mathbb{R}^n \to [0,1]$:

$$u^*(x) = \int_0^1 1_{\Sigma_{\mu,u*}}(x)\,\mathrm{d}\mu \tag{3.5}$$

we can rewrite the first term in the functional (3.2) as

$$\int_{\mathbb{R}^n} f u^* \,\mathrm{d}x = \int_{\mathbb{R}^n} f \left(\int_0^1 1_{\Sigma_{\mu,x}}\,\mathrm{d}\mu\right)\,\mathrm{d}x = \int_0^1 \int_{\Sigma_{\mu,u^*}} f(x)\,\mathrm{d}x \tag{3.6}$$

As a consequence, the functional (3.2) takes on the form:

$$E(u^*) = \int_0^1 \Big\{ \int_{\Sigma_{\mu,u^*}} f\,\mathrm{d}x \,+\, \big|\partial\Sigma_{\mu,u^*}\big|_g \Big\}\,\mathrm{d}\mu \;\equiv\; \int_0^1 \hat{E}\big(\Sigma_{\mu,u^*}\big)\,\mathrm{d}\mu, \tag{3.7}$$

where we have used the coarea formula to express the weighted total variation norm in (3.2) as the integral over the length of all level lines of u measured in the norm induced by g. Clearly the functional (3.7) is now merely an integral of the original binary labeling problem $\hat{E}$ applied to the upper level sets of u^*.

Assume that for some threshold value $\tilde{\mu} \in (0,1)$ theorem 1 was not true, i.e. there exists a minimizer Σ^* of the binary labeling problem with smaller energy:

$$\hat{E}(\Sigma^*) < \hat{E}(\Sigma_{\tilde{\mu},u^*}). \tag{3.8}$$

Then for the indicator function $1_\Sigma*$ of the set Σ^* we have:

$$E(1_\Sigma*) = \int_0^1 \hat{E}(\Sigma^*)\,\mathrm{d}\mu < \int_0^1 \hat{E}(\Sigma_{\mu,u^*})\,\mathrm{d}\mu = E(u^*), \tag{3.9}$$

which contradicts the assumption that u^* was a global minimizer of (3.2). □

Convex Minimization

Minimizing (3.2) is a constrained optimization problem due to the condition $0 \leq u \leq 1$. It can be transformed to an unconstrained one by dropping the constraint and adding a convex penalizer $\theta(u)$ to the energy. A suitable function for θ is [34]:

$$\theta(u) = \max\left\{0, 2\left|u - \tfrac{1}{2}\right| - 1\right\}. \tag{3.10}$$

This leads to the unconstrained convex minimization of the energy

$$E(u) = \int_{\mathbb{R}}^{n} f(x)\, u(x)\, \mathrm{d}x + \nu \int_{\mathbb{R}}^{n} g(x)\, |\nabla u(x)|\, \mathrm{d}x + \alpha \int_{\mathbb{R}}^{n} \theta(u(x))\, \mathrm{d}x \tag{3.11}$$

which has the same set of minimizers as (3.2) for a sufficiently large weighting parameter α.

In practice, the constraint $0 \leq u \leq 1$ is usually enforced by a respective projection (see Sec. 3.2.2).

Finding the Global Minimizer of the Original Binary Problem

Global minimizers of the functional (3.2) in a spatially continuous setting are therefore calculated as follows:

1. Compute a minimizer u^* of the energy (3.2) on the convex set of functions $u : \mathbb{R}^n \to \mathbb{R}$. Details are given in Sec. 3.2.2.

2. Threshold the minimizer u^* at some value $\mu \in (0, 1)$ to obtain a binary solution of the original shape optimization problem. Although these solutions generally depend on μ, all of them are guaranteed to be global minimizers of (3.2). In the experiments in this chapter it was set to $\mu = 0.5$.

3.2.2 Numerical Optimization

A minimizer of (3.2) must satisfy the Euler-Lagrange equation

$$0 = f(x) - \nu \operatorname{div}\left(g(x) \frac{\nabla u(x)}{|\nabla u(x)|}\right) \qquad \forall x \in \mathbb{R}^n. \tag{3.12}$$

Euler-Lagrange equations are used to compute a minimum of functionals of the following form:

$$E(u) = \int \mathcal{L}(u, \nabla u)\, \mathrm{dx}. \tag{3.13}$$

The function $\mathcal{L}(u, \nabla u)$ is called Lagrange function. A necessary condition for a minimum of E is that its corresponding Euler-Lagrange equation is fulfilled:

$$\frac{\partial E}{\partial u} = \frac{\partial \mathcal{L}}{\partial u} - \frac{d}{dx}\frac{\partial \mathcal{L}}{\partial \nabla u} = 0. \tag{3.14}$$

Solutions to this system of equations can be obtained by a variety of numerical solvers. Several popular methods of them will be discussed in the following.

Gradient Descent

Gradient descent methods are suitable to find minima of functionals that cannot be solved directly. They are based on a local optimization of an initialization along the negative gradient of the functional $E(u)$:

$$\frac{\partial u}{\partial t} = -\frac{\partial E}{\partial u}. \tag{3.15}$$

The gradient descent scheme for functional (3.13) is given by

$$\frac{\partial u}{\partial t} = -\frac{\partial E}{\partial u} = -\frac{\partial \mathcal{L}}{\partial u} + \frac{d}{dx}\frac{\partial \mathcal{L}}{\partial \nabla u}. \tag{3.16}$$

Iteration of the following update step yields a convergence of u to a local minimum of E:

$$u^{t+1} = u^t + dt\left(-\frac{\partial E}{\partial u^t}\right) \tag{3.17}$$

with time step dt. u^t is the value of u at iteration step t. The iterations can be stopped once the energy does not change any more. If the functional E is convex, the corresponding gradient descent scheme converges to a globally optimal solution.

The gradient descent scheme corresponding to (3.12) is derived from the Euler-Lagrange equation and yields

$$u^{t+1} = u^t + dt\left(-f(x) + \nu \operatorname{div}\left(g(x)\frac{\nabla u(x)}{|\nabla u(x)|}\right)\right). \tag{3.18}$$

The right hand side of (3.12) is the functional derivative of the energy (3.2) and gives rise to the gradient descent scheme while the gradient norm $|\nabla u(x)|$ needs to be replaced by a smoothed differentiable version

$$|\nabla u(x)|_\epsilon := \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \epsilon^2} \tag{3.19}$$

for a small value for $\epsilon > 0$.

In practice gradient descent methods are known to converge very slowly to the minimal energy.

Linearized Fixed-Point Iteration with SOR

Discretization of the Euler-Lagrange equation (3.12) leads to a sparse non-linear system of equations. This can be solved using a fixed point iteration scheme that transforms the non-linear system into a sequence of linear systems. These can be efficiently solved with iterative solvers, such as Jacobi, Gauss-Seidel, Successive over-relaxation (SOR), or multi-grid methods (also called FAS for "full approximation schemes").

Successive over-relaxation (SOR) is a generalization of the Gauss-Seidel method. The additional over-relaxation step yields faster convergence:

$$u^{t+1} = \omega \cdot \overline{u}^{t+1} + (1 - \omega) \cdot u^t, \tag{3.20}$$

with the Gauss-Seidel value $\overline{u}^{t+1}$ and a suitable choice for the over-relaxation parameter $\omega \in \mathbb{R}$. The method converges for $\omega \in (0, 2)$. The optimal value of ω depends on the linear system to be solved.

The only source of non-linearity in (3.12) is the diffusivity $d := \frac{g}{|\nabla u|}$. Starting with an (arbitrary) initialization, one alternates computing the diffusivities and solving the *linear* system of equations with fixed diffusivities. A corresponding SOR method as presented in [10] is given by the following update scheme: An update step for u at pixel i and time step k yields

$$u_i^{l,k+1} = (1 - \omega) u_i^{l,k} + \omega \frac{\nu \sum\limits_{j \in \mathcal{N}(i), j<i} g \cdot g_{i\sim j}^l u_j^{l,k+1} + \nu \sum\limits_{j \in \mathcal{N}(i), j>i} g \cdot g_{i\sim j}^l u_j^{l,k} - f_i}{\nu \sum\limits_{j \in \mathcal{N}(i)} g \cdot g_{i\sim j}^l}, \tag{3.21}$$

where $g_{i\sim j}^l$ is the diffusivity between pixel i and j, and $\mathcal{N}(i)$ is the 4-connected neighborhood around i. Empirically the fastest convergence rate was obtained for $\omega = 1.85$.

The additional constraint is fulfilled by projecting the current value of u to the set $[0, 1]$ after each iteration:

$$\pi_C(u) = \begin{cases} 1 & \text{if } u > 1 \\ u & \text{if } 0 \leq u \leq 1 \\ 0 & \text{if } u < 0 \end{cases} \tag{3.22}$$

which corresponds to a clipping of u to the range $[0, 1]$.

First Order Primal-Dual Optimization

Primal-dual optimization methods have established as an effective method for finding minimizers of functionals of the type of (3.2), because they typically yield fast convergence rates. The corresponding primal dual formulation [31] yields the following update steps for u:

$$\begin{aligned} p^{t+1} &= \pi_D(p^t + \tau_p \nabla u^t) && (3.23) \\ u^{t+1} &= u^t + \tau_u(\mathrm{div}(p^{t+1}) - f) && (3.24) \end{aligned}$$

with time steps $\tau_p, \tau_u \in \mathbb{R}$. A dual variable $p : \Omega \to \mathbb{R}^d$ is introduced which splits the optimization into a projected gradient descent of the primal variable u and a projected gradient ascent in the dual variable p.

The projection π_D is given by:

$$\pi_D(p) = \frac{p}{\max\left\{1, \frac{|p|}{g}\right\}}. \tag{3.25}$$

Convergence Criteria

The numerical schemes above consist of update steps for u converging to a solution of the energy functional. Hence, appropriate conditions are needed to determine when the optimization should be stopped. Using a fixed number of iterations is the most simple way, however choosing too many or too few iterations can unnecessarily increase the run time or yield solutions that are not sufficiently converged. Iterations usually can be stopped as soon as the energy decay in an iteration is lower than a given threshold. The convergence of the primal-dual optimization scheme can be estimated by its corresponding *primal-dual gap*, the difference between the primal and the dual energy.

3.2.3 Parallelization on Graphics Processing Units

PDE-based approaches are generally suitable for parallel computing on graphics cards: The gradient descent and primal-dual schemes are straightforward to parallelize. This does not hold for the standard Gauss-Seidel scheme as it requires sequential processing of the image. However, in its Red-Black variant the Gauss Seidel scheme is parallelizable. The same holds for its various derivates such as SOR and FAS. In this thesis, implementations were made using the GPU computing language *Cuda*.

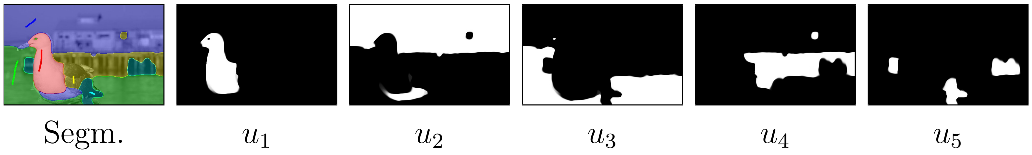


Figure 3.3: Image segmentation with five regions. Each region is assigned a labeling function $u_1, \dots, u_5$, whose values converge to binary, although optimized in $[0, 1]$. Results are shown before thresholding.

3.2.4 Multi-Label Segmentation

The more general formulation of the binary functional is the multi-label segmentation given by [119]:

$$\min_{u \in BV} \left\{ \sum_{i=1}^{n} \left(\int_\Omega f_i(I(x)) u_i \,\mathrm{d}x + \nu \int_\Omega |\nabla u_i| \,\mathrm{d}x \right) \right\}, \text{s.t.} \sum_{i=1}^{n} u_i = 1. \tag{3.26}$$

The additional constraint enforces $u(x)$ to lie on a simplex and implements the constraint that each pixels gets assigned exactly one label.

Primal-Dual Optimization

The following update scheme converges to an approximation of the minimum of (3.26) [119]:

$$p^{t+1} = \pi_D(p^t + \tau_p \nabla v^t) \tag{3.27}$$
$$u^{t+1} = \pi_B(u^t + \tau_u(\mathrm{div}(p^{t+1}) - f)) \tag{3.28}$$
$$v^{t+1} = 2u^{t+1} - u^t \tag{3.29}$$

The projection π_D is equivalent to (3.25):

$$\pi_D(p) = \frac{p}{\max\{1, |\nabla p|)\}}. \tag{3.30}$$

The projection onto the simplex π_B can be computed by the projection algorithm described in [106]. It ensures that the constraint that the labels should be disjunct is fulfilled, i.e. $\sum_{i=1}^{n} u_i = 1$. In this thesis the step sizes were set to $\tau = \sigma = 0.3$.

Fig. 3.3 shows an example image segmentation into five regions, as well as the five layers of u after convergence. As can be seen in the figure, the values of u converge to binary. The input image from Fig. 3.3 is from the IcgBenchmark [125]. A review of discrete and continuous methods for multi-label image segmentation methods can be found in [112].

3.3 Discrete Optimization with Graph Cuts

To solve the binary labeling problem (3.2) in a discrete setting, the input data is converted into a directed graph in form of a regular lattice: Each pixel (or voxel) in the input data corresponds to a node in the lattice. To approximate the metric g measuring the boundary size of the hypersurface S, neighboring nodes are connected. The degree of connectivity depends on the application. Details will be given in Sec. 3.3.2.

Additionally a source node s and a sink node t are introduced. They allow to include the unary terms $f(x)u(x)$ for the pixels x: If $f(x) \geq 0$, an edge to the source is introduced, weighted with $f(x)$. Otherwise an edge to the sink weighted with $-f(x)$ is created.

The optimal binary labeling u corresponds to the minimal s/t - cut in the graph. An s/t - cut is a partitioning of the nodes in the graph into two sets S and T, where S contains the source s and T the sink t. Nodes $x \in S$ are assigned the label $u(x) = 0$, nodes $x \in T$ the label $u(x) = 1$. The weight of such a cut is the sum of the weights of all edges starting in S and ending in T.

It was shown that an energy function E can be solved with graph cuts if it fulfills the *submodularity condition* [86] :

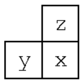
$$E(s,s) + E(t,t) \leq E(s,t) + E(t,s). \tag{3.31}$$

3.3.1 Computing Minimal Cuts in Graphs

Efficient solvers of the minimal s/t - cut problem are based on computing the maximal flow in the graph [51]. Such methods are divided into three major categories: those based on augmenting paths [51, 47, 24], blocking flows [46, 60] and the push-relabel method [59]. Some of these methods do not guarantee a polynomial running time [24] or require integral edge weights [60]. To solve 2-dimensional problems of form (3.2) usually the algorithm of Boykov and Kolmogorov performs best [24]. For highly connected three-dimensional grids the performance of this algorithm breaks down [24] and push-relabel methods become competitive. Recently efforts were made to parallelize push-relabel-based approaches [43].

3.3.2 Approximating Metrics using Graph Cuts

The question of how to approximate continuous metrics of the boundary size in a discrete setting has received significant attention by researchers. Boykov and Kolmogorov [23] show how to approximate any Riemannian metric, including anisotropic ones. In [84] they discuss how to integrate flux. A similar



gradient mask

$u(x)$	$u(y)$	$u(z)$	$\|\nabla u\|$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	$\sqrt{2}$

$u(x)$	$u(y)$	$u(z)$	$\|\nabla u\|$
1	0	0	$\sqrt{2}$
1	0	1	1
1	1	0	1
1	1	1	0

Figure 3.4: The L_2-norm of the 2D gradient as a ternary term. One easily verifies that this term is submodular.

construction can be derived from the divergence theorem. In the following the discussion is limited to the isotropic case.

The section starts with a review of the method in [23] which replaces the L_2-norm of the gradient in (3.2) by its L_1-norm. For the Euclidean metric ($g(x) = 1 \ \forall x \in \mathbb{R}^n$) then a novel discretization scheme is presented which allows to use the L_2-norm of the gradient based on higher order terms [5].

Approximation using Pairwise Terms

Based on the Cauchy-Crofton formula of integral geometry, Boykov and Kolmogorov [23] showed that the metric given by g can be approximated by connecting pixels to all pixels in a given neighborhood. The respective neighborhood systems can be expressed as

$$N_R(x) = \left\{ x + \left(\begin{array}{c} a \\ b \end{array} \right) \middle| a, b \in \mathbb{Z}, \, \sqrt{a^2 + b^2} \leq R, \, gcd(|a|, |b|) = 1 \right\} .$$

The constraint on the greatest common divisor avoids duplicate directions. The edge corresponding to $(a \ b)^\top$ is given a weight of $g(x)/\sqrt{a^2 + b^2}$. For $R = 1$ the obtained 4-connected lattice reflects the L_1-norm of the gradient. With increasing R and decreasing grid spacing the measure converges to the continuous measure. This is not true when fixing the connectivity (i.e. when keeping R constant).

Length Approximation using Higher Order Terms

This section presents a method to approximate length regularity in a graph cut based framework: Instead of using pairwise terms higher order terms are introduced. These allow to represent a more accurate discretization of the L_2-norm in the length term.

The energy (3.2) involves the L_2-norm of the generalized gradient of the $\{0, 1\}$-function u. With the pairwise terms discussed above a large connectivity is needed to approximate this norm. In the following, it will be shown that a more accurate approximation of the L_2-norm can be integrated in a graph cut framework, without increasing the connectivity. A central contribution of this chapter is to show how a more accurate approximation of the L_2-norm can be integrated in a graph cut framework. The key observation is that in a two-dimensional space a consistent calculation of the gradient is obtained by taking the differences to the upper and left neighbor in the grid – see Fig. 3.4.

Fig. 3.4 also shows the arising term. One easily verifies that this term satisfies the submodularity condition [86]. For a third order term as this one, this condition implies that the term can be minimized using graph cuts.

The corresponding term in 3D space where each pixel is connected to three neighbors was also considered. The arising fourth order term – with values in $\{0, 1, \sqrt{2}, \sqrt{3}\}$ – is submodular. However it is not clear whether it can be minimized via graph cuts: It does not satisfy the sufficient conditions pointed out by Freedman [53]. From a practical point of view, in 2D the novel terms do not perfom well: The length discretization only compares a pixel to those pixels in the direction of the upper left quadrant. Performance is boosted when adding the respective terms for the other three quadrants as well.

3.4 Comparison of Continuous and Discrete Shape Optimization

This section provides a quantitative comparison of spatially discrete and spatially continuous shape optimization schemes as presented in the previous two sections. While both approaches aim at minimizing the same functional, the following three important differences were identified in [5]:

Termination Criterion The spatially discrete approach has an exact termination criterion and a guaranteed polynomial running time (for a number of maximum-flow algorithms). On the other hand, the spatially continuous approach is based on the iterative minimization of a non-linear convex functional. While the required number of iterations is typically size-independent (leading to a computation time which is linear in the number of pixels/voxels), one cannot speak of a guaranteed polynomial time complexity. A termination criterion is needed to determine when the solution is converged.

Optimization Domain The spatially discrete approach is based on discretizing the cost functional on a lattice and minimizing the resulting submodular problem by means of graph cuts. The spatially continuous approach, on the other hand is based on minimizing the relaxed problem in a continuous setting where the resulting Euler-Lagrange equations are solved on a discrete lattice. This difference gives rise to metrication errors of the spatially discrete approach which will be discussed in Sec. 3.4.3.

Optimization Method The optimization of the spatially discrete approach is based on solving a maximum flow problem, whereas the spatially continuous approach is performed by solving a partial differential equation. This fundamental difference in the underlying computational machinery leads to differences in computation time, memory consumption and parallelization properties.

3.4.1 Computation Times

Numerous methods exist to solve either the discrete or the continuous optimization tasks. A comparison of all these methods is outside the scope of this chapter. Instead a few solvers are chosen that were considered competetive. For all graph cut methods the algorithm of [24] is used, which is arguably the most frequently used in Computer Vision. All discretizations mentioned above were tested.

For the TV segmentation sequential methods were implemented on the CPU and parallel solvers on a Geforce GTX 8800 graphics card using the CUDA framework. Both implementations are based on the SOR method. On the CPU the usual sequential order of pixels was used, and on the GPU the corresponding parallelizable Red-Black scheme where the image is divided according to a checkerboard pattern. A termination criterion is necessary as the number of required iterations depends on the length weight ν. We compare the segmentations every 50 iterations and stop as soon as the maximal absolute difference between two values of u drops below a value of 0.000125.

Computation Times for 2D Shape Optimization

Table 3.1 shows run-times for the mentioned methods. The task is image segmentation using the two-label piecewise constant Mumford-Shah with fixed mean values 0 and 1 on an intensity image with range $[0, 1]$. The main conclusions are summarized as follows:

- The TV segmentation profits significantly from parallel architectures. According to the results this is roughly a factor of 5. It should be noted

	Cameraman Image			Berkeley Arc Image		
	$\nu = 1$	$\nu = 3$	$\nu = 5$	$\nu = 1$	$\nu = 3$	$\nu = 5$
GC-4	0.02 s	0.1 s	0.33 s	0.06 s	0.16 s	0.53 s
GC-8	0.05 s	0.15 s	0.4 s	0.1 s	0.27 s	0.93 s
GC-16	0.2 s	0.35 s	0.95 s	0.33 s	0.85 s	2.7 s
TV-GD-CPU	111.38 s	251.97 s	259.87 s	409.08 s	636.28 s	157.64 s
TV-SOR-CPU	10.9 s	13.26 s	10.2 s	35.89 s	103.5 s	39.26 s
TV-SOR-GPU	2 s	2.7 s	2 s	7.6 s	28.3 s	8.6 s
Acc. factor	5.45	4.91	5.1	4.72	3.66	4.57

Table 3.1: 2D image segmentation: Run-times for the different optimization methods on two different images. The following methods were compared: Graph Cuts 4-connected (GC-4), Graph Cuts 8-connected (GC-8), Graph Cuts 16-connected (GC-16), TV with gradient descent on CPU (TV-GD-CPU), TV with SOR on CPU (TV-SOR-CPU) and TV with red-black SOR on GPU (TV-SOR-GPU). The last row shows the acceleration factor of Red-black TV-SOR on a GPU compared to TV-SOR on a CPU.

Graph cuts 6-connected	13 s
Graph cuts 26-connected	12 min 35 s
TV with SOR (CPU)	9 min 36 s
TV with red-black SOR (GPU)	30 s

Table 3.2: Run-times for the 3D catenoid example shown in Fig. 3.8.

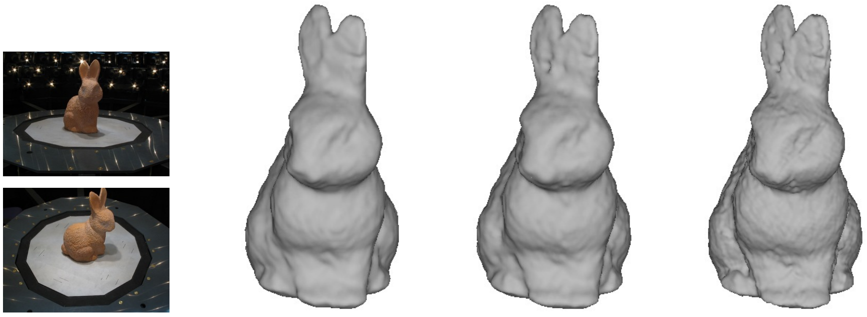
that the GPU-implementation usually requires more iterations because the Red-Black order is used.

- The graph cut based methods clearly outperform the TV segmentation, even on the GPU.

- While for the graph cut methods the 16-connected pairwise terms give generally the best results (they are largely free from grid bias), they also use up the most run-time.

Computation Times for 3D Shape Optimization

For 3D shape optimization the connectivity of the graph in discrete formulations plays a more important role. While the 4-connected pixel grid in 2D corresponds to a 6-connected voxel grid, the inclusion of diagonal edges



Two of 33 input images | Graph Cuts (108×144×162) | Convex TV (108×144×162) | Convex TV (216×288×324)

Figure 3.5: Comparison of discrete and continuous optimization for multiview 3D reconstruction (presented in [10]): Due to the dominant data fidelity term, the discrete and continuous reconstructions are similar for the same volume resolution. However, for increasing resolution more accurate results can be achieved with the continuous formulation, while graph cuts rapidly come across memory limitations.

corresponds to an 8-connected grid in 2D and already a 26-connected grid in 3D and has thus a much higher impact on the computation times.

Table 3.2 shows run-times of the different optimization methods for the 3D catenoid example shown in Fig. 3.8. Three main conclusions were detected:

- The 6-connected graph cuts method is the fastest, however it computes the wrong solution (see Fig. 3.8).
- The run-time of the graph cut method changes for the worse with higher connectivity, and gets slower than the TV optimization, both on CPU and GPU. Note that this limitation is due to the fact that the Boykov-Kolmogorov algorithm [24] is optimized for sparse graph structures. For denser (3D) graphs alternative push-relabel algorithms might be faster.
- The parallel implementation of the TV method allows for a speed up factor of about 20 compared to the CPU version.

3.4.2 Memory Consumption

With respect to the memory consumption the TV segmentation is the clear winner: It requires only one floating point value for each pixel in the image.

In contrast, graph cut methods require an explicit storage of edges as well as one flow value for each edge. The number of edges is here dependent on the connectivity. This difference becomes especially important for high resolutions in 3D, as can be seen in the experiment in Fig. 3.5. Increasing the volume resolution to $216\times288\times324$ yields more accurate results for the continuous TV formulation, while a graph cut solution for this resolution was not feasible to compute because of memory limitations, even on the 6-connected grid.

3.4.3 Metrication Errors and Consistency

Although both approches minimize the same function, the discrete approach tends to prefer the directions of the underlying grids while the continuous TV segmentation shows no such preference. The continuous formulation allows for a consistent discretization. This means that the solutions converge to the continuous solution for increasing grid resolutions. The continuous formulation of the relaxed problem allows for arbitrarily accurate results in the range of subpixels. Therefore the continuous approach does not suffer from discretization artefacts like the discrete graph cuts approach, which is based on a discretization of the energy.

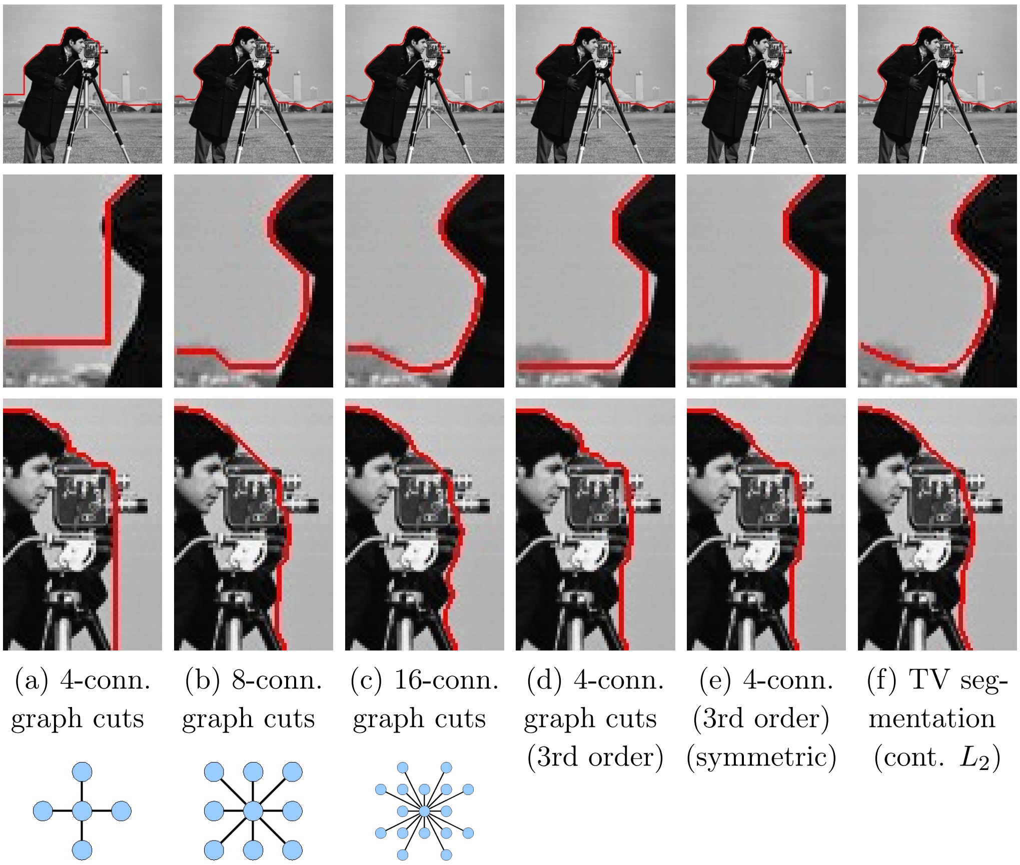
Accuracy Comparison in 2D for Image Segmentation

Fig. 3.6 shows a comparison of graph cut approaches with the continuous total variation (TV) segmentation. It shows several ways to deal with the discretization of the metric for graph cuts are shown. None of the graph cut approaches produces such a smooth curve as the TV segmentation, although the 16-connected grid gets quite close to it. This inspired us to investigate the source for the metrication errors arising in graph cut methods.

On the 4-connected grid in $\mathbb{R}^2$, for example, graph cuts usually approximate the Euclidean boundary length of the interface S as

$$|S| = \int\limits_S dS \approx \frac{1}{2}\sum_i \sum_{j\in\mathcal{N}(i)} |u_i - u_j|, \tag{3.32}$$

where $\mathcal{N}(i)$ denotes the four neighbors of pixel i. This implies that the boundary length is measured in an L_1-norm rather than the L_2-norm corresponding to the Euclidean length. The L_1 norm clearly depends on the choice of the underlying grid and is not rotationally invariant. Points of constant distance in this norm form a diamond rather than a circle (see Fig. 3.7). This leads to a preference of boundaries along the axes (see Fig. 3.6(a)).



(a) 4-conn. graph cuts

(b) 8-conn. graph cuts

(c) 16-conn. graph cuts

(d) 4-conn. graph cuts (3rd order)

(e) 4-conn. (3rd order) (symmetric)

(f) TV segmentation (cont. L_2)

(g) 4-conn. grid

(h) 8-conn. grid

(i) 16-conn. grid

Figure 3.6: Comparison of different norms and neighborhood connectivities for discrete (a-e) and continuous (f) optimization for image segmentation. For the discrete solution a 16-connected graph (c) is necessary to obtain similiar results to the continuous solution (f). Increasing the connectivity of the grid (g)–(i) reduces metrication errors but can lead to memory limitations.

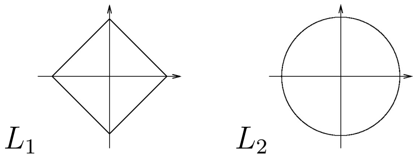


Figure 3.7: 2D visualization of the L_1-norm and the L_2-norm for points of constant distance: Unlike the L_1-norm, the L_2-norm is rotationally invariant.

This dependency on the underlying grid can be reduced by increasing the neighborhood connectivity. By reverting to larger and larger neighborhoods one can gradually eliminate the metrication error [23]. Increasing the connectivity leads in fact to better and better approximations of the Euclidean L_2-norm (see Fig. 3.6(b) and 3.6(c)).

Yet, a computationally efficient solution to the labeling problem requires to fix a choice of connectivity. And for any such choice, one can show that the metrication error persists, that the numerical scheme is not *consistent* in the sense that a certain residual reconstruction error (with respect to the ground truth) remains and cannot be eliminated by increasing the resolution.

Since the spatially continuous formulation is based on a representation of the boundary length by the L_2-norm:

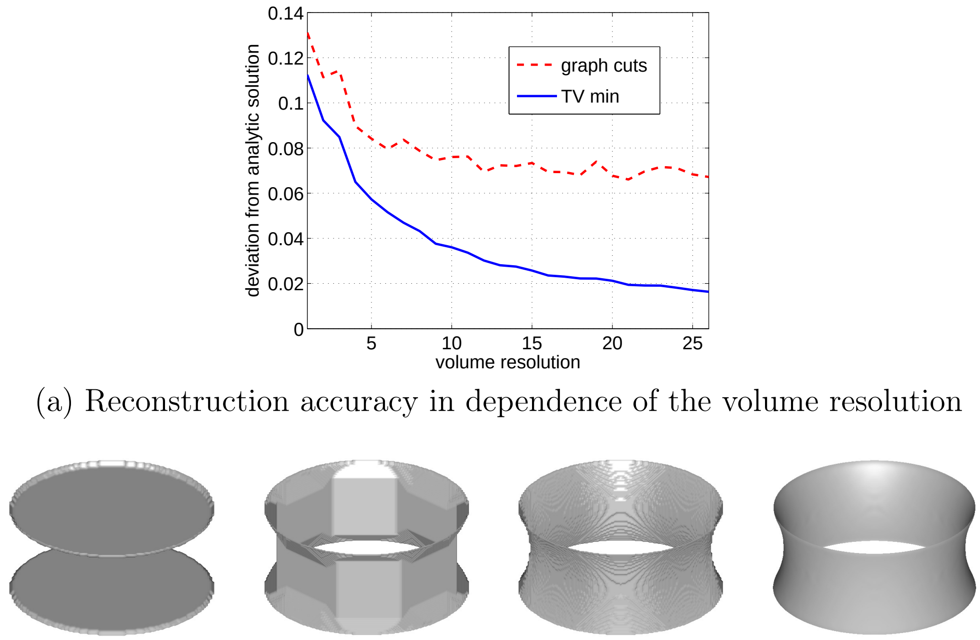
$$|S| = \int\limits_S \mathrm{d}S = \int\limits_\Omega |\nabla u| \,\mathrm{d}x = \int\limits_\Omega \sqrt{u_x^2 + u_y^2}\,\mathrm{d}x, \tag{3.33}$$

the resulting continuous numerical scheme does not exhibit such metrication errors (see Fig. 3.6(f)). The TV segmentation performs optimization in the convex set of functions with range in $[0,1]$. It hence allows intermediate values where the graph cut only allows binary values.

The third order graph cuts discretization of the L_2-norm (see Fig. 3.6(d) and 3.6(e)) computes the same discretization of the L_2-norm, however allowing only for binary values. Hence, in this discretized version, the Euclidean length is computed for angles of 45° and 90° to the grid, by using only a 4-connected grid. Therefore the third order L_2-norm leads to similar results on a 4-connected grid as second order terms on an 8-connected grid.

Accuracy Comparison in 3D for a Catenoid

Fig. 3.8 shows a synthetic experiment of solving a minimal surface problem with given boundary constraints using the example of a bounded catenoid. As the true solution of this problem can be computed analytically, it is suitable for a comparison of different solvers. The experiment compares graph cuts



(a) Reconstruction accuracy in dependence of the volume resolution

(b) 6-connected Graph cuts (L_1)

(c) 26-connected Graph cuts

(d) Convex TV (L_2)

(e) Analytic Solution

Figure 3.8: Comparison of discrete and continuous optimization methods for the reconstruction of a catenoid: While the discrete graph cut algorithm exhibits prominent metrication errors (polyhedral structures), the continuous method does not show these. The plot shows the accuracy of the 26-connected graph cuts and the continuous TV method in dependence of the volume resolution. The consistency of the continuous solution is validated experimentically in the sense that the reconstruction error goes to zero with increasing resolution.

and continuous TV minimization. It demonstrates that the 6-neighborhood graph cuts method completely fails to reconstruct the correct surface topology – in contrast to the full 26-neighborhood which approximates the Euclidean metric in a better way. However, discretization artifacts are still visible in terms of polyhedral blocky structures. Fig. 3.8 also shows the deviation of the computed catenoid solutions from the analytic ground-truth for increasing volume resolution. It shows that for a fixed connectivity structure the computed graph cut solution is not consistent with respect to the volume resolution. In contrast, for the solution of the continuous TV minimization the discretization error decays to zero.

The experiment demonstrates that graph cut solutions can indeed be improved by reverting to larger neighborhood connectivity (26 instead of 6 neighbors). Yet, for any connectivity there is a metrication error, which persists with increasing resolution. The continuous TV optimization, on the other hand, is consistent as the discretization error decays to zero.

Fig. 3.5 shows an experiment for real image data. In this multiview reconstruction problem the data fidelity term is dominant, therefore the discrete and the continuous solutions are similar for the same volume resolution ($108\times144\times162$).

3.5 Conclusion

A certain class of shape optimization functionals can be globally minimized both in a spatially discrete and in a spatially continuous setting. This chapter reviewed these recent developments and presented an experimental comparison of the two approaches regarding the accuracy of reconstructed shapes and computational speed and memory requirements.

This chapter described how convex relaxation methods allow for global optimization of the two-label piecewise constant Mumford-Shah functional, in addition to its discrete formulation with graph cuts.

A detailed quantitative analysis of the presented continuous and discrete shape optimization methods confirmed the following differences:

- Spatially discrete approaches generally suffer from metrication errors in the approximation of geometric quantities such as boundary length or surface area. These arise due to the binary optimization on a discrete lattice. These errors can be alleviated by reverting to larger connectivity. Alternatively, it was shown that higher-order terms allow to implement an L_2-norm of the gradient, thereby providing better spatial consistency without extending the neighborhood connectivity. As the spatially continuous formulation is not based on a discretization of the cost functional but rather a discretization of the numerical optimization (using real-valued variables), it does not exhibit metrication errors in the sense that the reconstruction errors decay to zero as the resolution is increased.

- The spatially continuous formulation allows for a straight-forward parallelization of the partial differential equation. As a consequence, one may obtain lower computation times compared to respective graph cut methods, in particular for the denser graph structures prevalent in 3D shape optimization.

- While the discrete graph cut optimization can be performed in guaranteed polynomial time, this is not the case for the analogous continuous shape optimization. While respective termination criteria for the convex optimization work well in practice, defining termination criteria that apply to any shape optimization problem remains an open problem.

For the methods in the subsequent chapters, continuous formulations were chosen because of their better performance concerning memory requirement, the ability of parallelization, and the avoidance of metrication errors.

Chapter 4

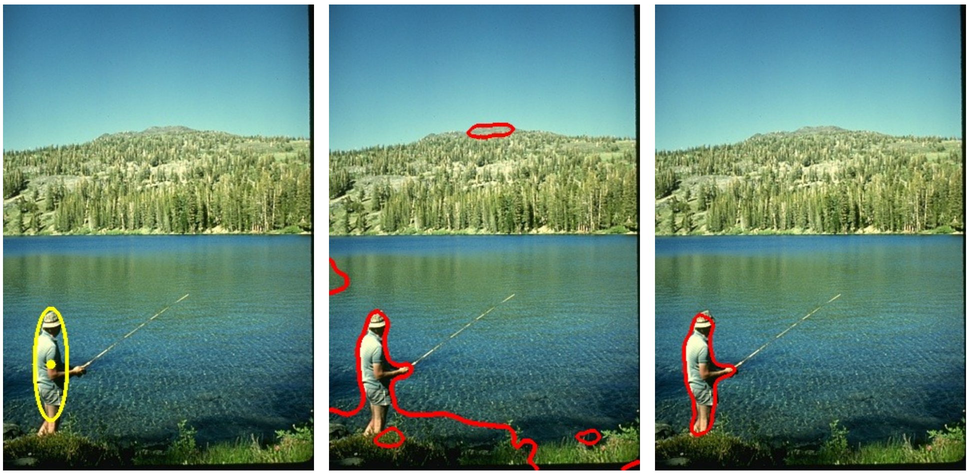
Image Segmentation with Moment Constraints

Prior knowledge about objects can be helpful especially in difficult segmentation tasks. This chapter shows how convex shape constraints based on the lower order moments of a shape can be integrated into convex relaxation methods. Shape priors in terms of moment constraints can be imposed within the convex optimization framework, since they give rise to convex constraints and therefore allow for global optimization. The chapter will focus on the lower order moments of shapes, which correspond to the area or volume, the centroid, and the variance or covariance. Constraints on these lower order moments can be intuitively imposed in an interactive user interface or deduced from arbitrary shapes. Respective constraints can be imposed as hard constraints or soft constraints. Quantitative experiments on a variety of images demonstrate that the user can impose such constraints with a few mouse clicks, leading to substantial improvements of the resulting segmentation, and reducing the average segmentation error from 12% to 0.35%. GPU-based computation times of around 1 second allow for interactive applications. Furthermore, an extension of the method to object tracking in image sequences based on moment constraints will be shown.

Parts of this chapter have been published in [3] and [6].

4.1 Introduction

Imposing shape constraints for image segmentation is an established method to incorporate prior knowlege about the objects to segment into the optimization.



(a) User input

(b) Color only Segmentation

(c) with Moment Constraints

Figure 4.1: Interactive image segmentation with constraints on the lower order moments of a shape. Constraints on the area, centroid and covariance are easily transmitted through mouse interaction (left). They allow to stabilize the segmentation process while preserving fine-scale details of the shape (right).

This chapter is focussed on functionals of the form:

$$E(S) = \int\limits_{int(S)} f(x)\,\mathrm{d}x \; + \int\limits_{S} g(x)\,\mathrm{dA}, \tag{4.1}$$

where S denotes a hyper surface in $\mathbb{R}^d$, i.e. a set of closed boundaries in the case of $2D$ image segmentation or a set of closed surfaces in the case of $3D$ segmentation and multi view reconstruction. The functions $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}^+$ are application dependent. In a statistical framework for image segmentation, for example,

$$f(x) = \log p_{bg}(I(x)) - \log p_{ob}(I(x)), \tag{4.2}$$

may denote the log likelihood ratio for observing the color $I(x)$ at a point x given that x is part of the background or the object, respectively.

The second term in (4.1) corresponds to the area (for $d = 3$) or the boundary length (for $d = 2$), measured in a metric given by the function g. In the context of image segmentation, g may be a measure of the local edge strength – as in the geodesic active contours [30, 79] – which energetically favors segmentation boundaries along strong intensity gradients. In the context of multi view reconstruction, $g(x)$ typically measures the photo-consistency among different views of the voxel x, where low values of g indicate a strong agreement from different cameras on the observed patch intensity.

This chapter shows that one can impose moment constraints of arbitrary order in the framework of convex shape optimization, thereby generalizing from the zeroth order moment (area/volume) to higher order moments (centroid, scale, covariance, etc). In particular, all moment constraints – both soft and hard – correspond to convex constraints. As a consequence we can compute moment-constrained shapes which are independent of initialization and lie within a bound of the optimum.

The outline of this chapter is as follows. Sec. 4.2 briefly reviews related work on shape priors for segmentation. Sec. 4.3 shows that moment constraints can be imposed as convex constraints within convex relaxation optimization. Sec. 4.4 shows how the arising optimization problem can be minimized using efficient GPU-accelerated PDE solving. Furthermore it is shown that computing projections onto the moment constraint sets can be efficiently computed by solving systems of linear equations. Sec. 4.5 presents a variant of the proposed method using ratio constraints. Sec. 4.6 presents experimental results and a quantitative evaluation showing that interactive segmentation results can be drastically improved using moment constraints. Sec. 4.7 shows how the presented method for image segmentation with moment constraints can be extended to a method for object tracking in videos.

4.2 Related Work

There has been much research on imposing prior shape knowledge into image segmentation. It was shown that segmentation results can be substantially improved by imposing shape priors [68, 39, 52].

Recent approaches are able to compute globally optimal solutions for segmentation problems with shape priors. In [128] a combinatorial solution for imposing shape priors to image segmentation based on the product graph was presented. An interesting property of the method is the rotational invariance in addition to the translational invariance which is achieved by a sampling of the rotation space. However, the method is based on learning of reference shape making it less general.

Approaches for shape constraints without a previous learning of reference shapes include convex formulations for connectivity constraints [135] and graph cut segmentation of compact objects [41] and star-shaped objects [143]. A graph based approach for shape constraints on sizes in segmentation was presented in [96]. In [63] a shape prior for convexity in shapes for image segmentation was presented, however no guarantee can be made that the results are global optmimal solutions.

Many shape priors have a rather fine granularity in the sense that they impose the object silhouette to be consistent with those silhouettes observed in a training set [39, 48]. The degree of abstraction is typically rather small. In particular, deviations of the observed shape from the training shapes are (elastically) suppressed by the shape prior. This is particularly undesirable in medical image segmentation where malformations of organs (that make it deviate from the training shapes of healthy organs) should be detected rather than ignored. Other examples include natural objects where different specimen inherently do not have exactly the same shape, like leaves or animals. It may therefore be of interest to merely impose some coarse-level shape information rather that imposing the exact form of the object.

An alternative approach that may provide a remedy for the above problems is to impose moment constraints. In particular, the lower-order moments allow to constrain the area/volume, the centroid and the size or covariance of objects without imposing any constraints on their local shape. A related idea of using *Legendre moments* (albeit in a local optimization scheme) was developed in [52].

In a convex formulation of multiple view 3D reconstruction, it was shown that one can impose additional convex constraints which assure that the computed minimal surfaces are silhouette-consistent [82]. Essentially this constraint can be seen as a *volume constraint*: The volume along any ray from the camera center must be at least 1 if that ray passes through the silhouette and zero otherwise. In the two-dimensional case, a related constraint was proposed as a bounding box prior for image segmentation [95].

4.3 Moment Constraints for Image Segmentation

Functionals of the form (4.1) can be globally optimized in a spatially continuous setting by means of convex relaxation and thresholding [34]. To this end, one reverts to an implicit representation of the hyper surface S using an indicator function $u \in BV(\mathbb{R}^d; \{0,1\})$ on the space of binary functions of bounded variation, where $u = 1$ and $u = 0$ denote the interior and exterior of S. The functional (4.1) defined on the space of surfaces S is therefore equivalent to the functional

$$E(u) = \int_\Omega f(x)\, u(x)\, \mathrm{dx} + \int_\Omega g(x)|Du(x)|, \tag{4.3}$$

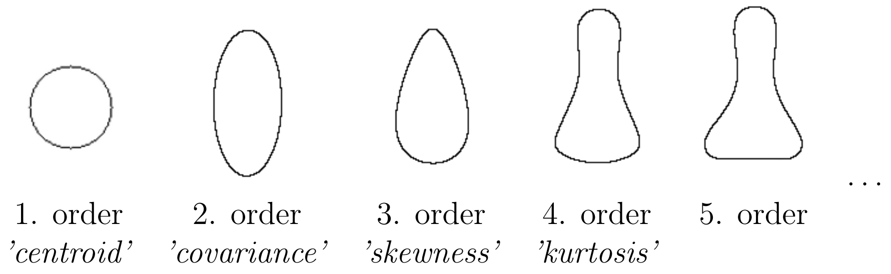


Figure 4.2: 2D central moments of a shape. With increasing order of the moment more details of the shape can be described.

where the second term in (4.3) is the weighted total variation. Here Du denotes the distributional derivative which for differentiable functions u boils down to $Du(x) = \nabla u(x)\mathrm{d}x$. By relaxing the binary constraint and allowing the function u to take on values in the interval between 0 and 1, the optimization problem becomes that of minimizing the convex functional (4.3) over the convex set $BV(\mathbb{R}^d; [0,1])$. Global minimizers u^* of this relaxed problem can therefore be computed, for example by a gradient descent procedure.

The thresholding theorem [34] assures that thresholding the solution u^* of the relaxed problem preserves global optimality for the original binary labelling problem.

In the following it will be shown that the moments of a segmentation can be successively constrained. These constraints give rise to nested convex sets. To this end shapes in d dimensions will be represented as binary indicator functions $u \in BV(\Omega; \{0,1\})$ of bounded variation on the domain $\Omega \subset \mathbb{R}^d$. We will denote the convex hull of this set by $\mathcal{B} = BV(\Omega; [0,1])$.

Fig. 4.2 shows examples for the first five 2D central moments of a shape. The first order moment describes the centroid of a shape, the second describes the covariance, i.e. the relation between width and height of an object. The lower order moments are intuitively the area, centroid and covariance dimensions of a shape. While egg-shapes of shapes can be described with the third order moment, for the fourth order moments the effect on the shape already becomes less intuitive. The figure shows that the higher the order of a moment, the more sophisticated properties can be imposed on the corresponding shape.

The following sections will show how these moments can be used to constrain properties of shapes for segmentation in a convex framework, and how especially the lower order moments can constrain shape optimization in an intuitive way.

4.3.1 Area Constraint

The area constraint arises from the 0th order moment is a measure for the mass of a shape. The area of the shape u can be constrained to be bounded by a constant $c \in \mathbb{R}^+$ by constraining u to lie in the set:

$$\mathcal{C}_0 = \left\{ u \in \mathcal{B} \,\middle|\, \int_\Omega u \,\mathrm{d}x = c \right\}. \tag{4.4}$$

In the case of a 3 dimensional domain Ω the constant c constrains the volume of a shape.

Proposition 1. *For any constant $c \geq 0$, the set $\mathcal{C}_0$ is convex.*

Proof. Let $u_1, u_2 \in \mathcal{C}_0$ be two elements from this set. Then for any $\alpha \in [0,1]$ the following holds:

$$\int_\Omega \alpha u_1 + (1-\alpha) u_2 \,\mathrm{d}x = \alpha \int_\Omega u_1 \,\mathrm{d}x + (1-\alpha) \int_\Omega u_2 \,\mathrm{d}x. \tag{4.5}$$

As a consequence the convex combination $u_\alpha := \alpha u_1 + (1-\alpha) u_2$ has the area $\int_\Omega u_\alpha \,\mathrm{d}x = c$ such that $u_\alpha \in \mathcal{C}_0$. □

In practice, we can either impose an exact area or we can impose upper and lower bounds on the area with two constants $c_1 \leq c_2$ and constrain the area to lie in the range $[c_1, c_2]$.

Alternatively, one can impose a soft area constraint by enhancing the functional (4.3) as follows:

$$E_0(u, \lambda_0) = E(u) + \lambda_0 \left(\int_\Omega u \,\mathrm{d}x - c \right)^2, \tag{4.6}$$

which imposes a soft constraint with a weight $\lambda_0 > 0$ favoring the area of the estimated shape to be near $c \geq 0$. The functional (4.6) is also convex.

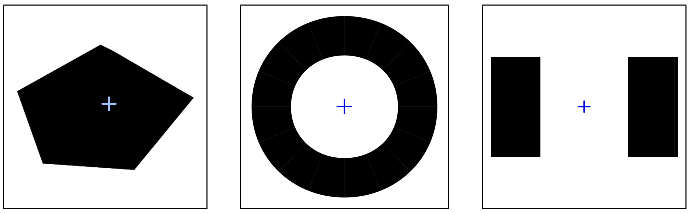
4.3.2 Centroid Constraint

The first order moment gives rise to the centroid constraint. In statistical measurement the first order moment corresponds to the mean of a probability density. Given the bounds about the centroid (center of gravity) for the object to be reconstructed, the centroid of the shape can be constrained by constraining the solution u to the set $\mathcal{C}_1$:

$$\mathcal{C}_1 = \left\{ u \in \mathcal{B} \,\middle|\, \frac{\int_\Omega x u \,\mathrm{d}x}{\int_\Omega u \,\mathrm{d}x} = \mu \right\}, \tag{4.7}$$

where equality is to be taken point wise and $\mu \in \mathbb{R}^d$. Alternatively, we can impose the centroid to lie between two constants $\mu_1, \mu_2 \in \mathbb{R}^d$. For $\mu_1 = \mu_2$, the centroid is fixed. $\mathcal{C}_1$ contains all shapes whose first order moment is μ. In probability theory the first order moment is also called the expected value.

Note that the centroid does not necessarily need to be located inside the object, since no connectivity or topological constraints are imposed on the segmentation. This can be seen in the ring shape and the two region shape depicted in Fig. 4.3 (b) and (c). Of the three shown examples, only the centroid of the convex shape (Fig. 4.3 (a)) is inside the object.



(a) Convex (b) Non-convex (c) Two parts

Figure 4.3: The centroid of a shape is not necessarily part of the object itself (b) and (c). Here, this only holds for the convex shape (a).

Proposition 2. *For any constant $\mu \geq 0$, the set $\mathcal{C}_1$ is convex.*

Proof. The equality constraint in (4.7) is equivalent to

$$\int_\Omega xu \,\mathrm{d}x = \mu \int_\Omega u \,\mathrm{d}x, \tag{4.8}$$

which is clearly a linear constraint. □

Reformulating the constraint equation in (4.7) yields the linear equation

$$\int_\Omega (\mu - x)u \,\mathrm{d}x = 0. \tag{4.9}$$

This allows to impose the centroid as a soft constraint by minimizing the energy:

$$E_1(u, \lambda_1) = E(u) + \lambda_1 \left(\int_\Omega (\mu - x)u \,\mathrm{d}x \right)^2. \tag{4.10}$$

Proposition 3. *Energy (4.10) is also convex in u.*

Proof. The first term in (4.10), $E(u)$ is convex, therefore the following will prove the convexity of the second term in (4.10), denoted as $\tilde{E}_1$. The convexity of $\tilde{E}_1$ can be shown using the definition of convex functions. For any $u_1, u_2 \in \mathcal{C}_1$ that fulfill the centroid constraint the following holds for all $\alpha \in [0,1]$:

$$\begin{aligned}
&\tilde{E}_1(\alpha u_1 + (1-\alpha)u_2, \lambda_1)\\
&= \lambda_1 \left(\int_\Omega (\mu - x)(\alpha u_1 + (1-\alpha)u_2) \, \mathrm{d}x \right)^2\\
&= \lambda_1 \left(\alpha \int_\Omega (\mu - x) u_1 \, \mathrm{d}x + (1-\alpha) \int_\Omega (\mu - x) u_2 \, \mathrm{d}x \right)^2\\
&\leq \alpha\lambda_1 \left(\int_\Omega (\mu - x) u_1 \, \mathrm{d}x \right)^2 + (1-\alpha)\lambda_1 \left(\int_\Omega (\mu - x) u_2 \, \mathrm{d}x \right)^2\\
&= \alpha \tilde{E}_1(u_1, \lambda_1) + (1-\alpha)\tilde{E}_1(u_2, \lambda_1)
\end{aligned}$$

□

4.3.3 Covariance Constraint

In the following, the concept will be generalized to moments of successively higher order, while the focus is the central moments, i.e. moments with respect to a specified centroid. In particular, the respective constraint structures are tensors of a dimension corresponding to the order of the moment.

The covariance structure is given by the second order moment. It can be imposed by constraining u to lie in the following convex set:

$$\mathcal{C}_2 = \left\{ u \in \mathcal{B} \;\middle|\; \frac{\int_\Omega (x-\mu)(x-\mu)^\top u \, \mathrm{d}x}{\int_\Omega u \, \mathrm{d}x} = A \right\}, \tag{4.11}$$

where the equality constraint should be taken element wise. Here $\mu \in \mathbb{R}^d$ denotes the center and $A \in \mathbb{R}^{d \times d}$ denotes a symmetric matrix whose elements are the constraint parameters. Alternatively, we can impose the covariance to lie in a range between two symmetric matrices $A_1, A_2 \in \mathbb{R}^{d \times d}$ such that $A_1 \leq A_2$ element-wise. Constraining the covariance, i.e. the second order moment, we are able to constrain the relation between width and height of an object. This constraint is particularly meaningful if one additionally constrains the centroid to be μ, i.e. considers the intersection of the set $\mathcal{C}_2$ (4.11) with a set of the form $\mathcal{C}_1$ (4.7).

Proposition 4. *For any constant $A \geq 0$, the set $\mathcal{C}_2$ is convex.*

Proof. The proof is analogous to that of proposition 2. □

We can derive the corresponding soft constraint on the covariance matrix by adding a respective term to the original energy:

$$E_2(u, \lambda_2) = E(u) + \lambda_2 \left(\int_\Omega \left(A - (x-\mu)(x-\mu)^\top \right) u \, \mathrm{d}x \right)^2 . \tag{4.12}$$

This functional is also convex, which can be proved analogously to the proof of Proposition 3.

Note that this allows, in particular, to constrain the scale σ of the object, because:

$$\sigma^2 = \frac{\int_\Omega |x-\mu|^2 u \, \mathrm{dx}}{\int_\Omega u \, \mathrm{d}x} = \mathrm{tr} \frac{\int_\Omega (x-\mu)(x-\mu)^\top u \, \mathrm{d}x}{\int_\Omega u \, \mathrm{d}x} . \tag{4.13}$$

From the constraint in (4.11) it follows that:

$$\mathrm{tr}(A_1) \leq \sigma^2 \leq \mathrm{tr}(A_2), \tag{4.14}$$

where tr denotes the trace of a matrix.

4.3.4 Higher Order Moment Constraints

In more general terms, the respective constraint set for moments of any order $k \in \mathbb{N}$ is given by:

$$\mathcal{C}_k = \left\{ u \in \mathcal{B} \;\middle|\; \frac{\int_\Omega (x_1-\mu_1)^{i_1} \cdots (x_d-\mu_d)^{i_d} u \, \mathrm{d}x}{\int_\Omega u \, \mathrm{d}x} = a_{i_1..i_d} \right\} , \tag{4.15}$$

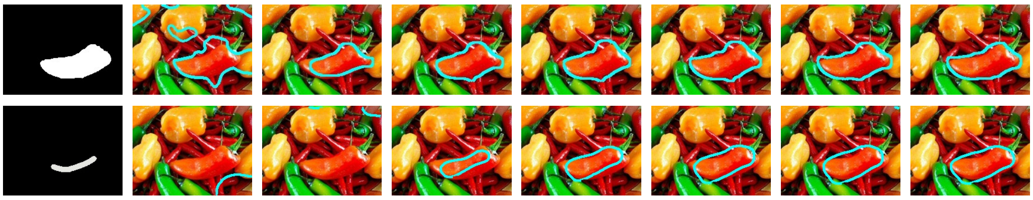
where $i_1 + \cdots + i_d = k$ and $a_{i_1..i_d}$ can be chosen appropriately to constrain the moment tensor of order k. Here x_i and μ_i denotes the i-th component of x and μ respectively.

Proposition 5. *For all $i_1, \ldots, i_d \in \mathbb{N}$ and for any constants $a_{i_1..i_d} \leq b_{i_1..i_d}$, the set $\mathcal{C}_{i_1 \ldots i_d}$ is convex.*

Proof. The proof is analogous to that of proposition 2. □

A Hierarchy of Shape Details

The above properties allow to impose various constraints on the shape associated with the indicator function u. Imposing more and more constraints of increasingly higher order leads to a decreasing intersection of the associated convex sets as a feasible domain of the shape and a corresponding hierarchy of shape details in segmentations. How much shape detail can one impose in this manner?



(a) Input Shape (b) no constr. (c) 0th order (d) up to 1st (e) up to 2nd (f) up to 3rd (g) up to 6th (h) up to 12th

Figure 4.4: Segmentation results with higher order moment constraints: By imposing constraints of increasing order (up to 12th order) more and more fine scale details of the shape are restored. For higher order moments the shape constraints can be derived either from reference shapes (first row) or user scribbles (second row).

Proposition 6. *Similarity to any given shape can be imposed at arbitrary detail by imposing convex moment constraints of increasingly higher order.*

Proof. According to the uniqueness theorem of moments [114], the function u is uniquely defined by its moment sequence. □

Fig. 4.4 shows an example of segmentations with high order moment constraints: While the higher-order moments allow to recover fine-scale shape details, the shape improvements due to higher order constraints are fairly small. Furthermore imposing moments of higher order is not very practical: Firstly, the user cannot estimate these moments visually. Secondly, the user cannot transmit respective higher-order tensors through a simple mouse interaction. Instead, having the image data determine the shape's fine scale structure turns out to be far more useful. Fig. 4.4 (a) shows the manually segmented input shape which is used to compute color histograms for foreground and background, as well as the moments that constrain the subsequent image segmentation. For this image the segmentation based on color histograms for foreground and background only is not sufficient to segment the red pepper from the other ones (Fig. 4.4 (b)). The first three moment constraints – area, centroid and covariance – are able to substantially improve segmentations (Fig. 4.4 (c)-(e)). With increasing order of moments more and more fine scale details of the shape can be reconstructed (Fig. 4.4 (f)-(h)). Even the 12th order moment constraint still results in an improvement of segmentation (Fig. 4.4 (h)). In the first row of the example shown in Fig. 4.4 the respective higher order moment constraints have been learned from a reference shape. For interactive applications however the higher order moments are too less intuitive to be applicable. The second row shows moment constraints derived from manually marked user scribbles. The resulting segmentations are less accurate compared to learning the constraints from reference shape. Again, segmentations improve with increasing order of the moments. In this example, the constraint parameters were derived from the user scribble except the area constraint which was estimated additionally.

4.4 Optimization with Moment Constraints

Shape optimization and image segmentation with respective moment constraints can now be done by minimizing convex energies under respective convex constraints.

Let $\mathcal{C}$ be a specific convex set containing knowledge about respective moments of the desired shape – given by an intersection of the above convex sets. Then we can compute segmentations by solving the convex optimization problem

$$\min_{u \in \mathcal{C}} E(u), \tag{4.16}$$

with $E(u)$ given in (4.3). The respective Euler-Lagrange equation is given by:

$$0 = \operatorname{div}\left(g \frac{\nabla u}{|\nabla u|} \right) - f. \tag{4.17}$$

The equation system can be solved using gradient descent, or the lagged diffusivity approach that was presented in [9]. We use the latter because in our experiments it achieves a speed up of computation times of a factor of ~ 5.

In the case of segmentation without moment constraints, u has to fulfill the constraint that $u \in \mathcal{B}$. It can be enforced by clipping u to the range $[0, 1]$ after each iteration of the optimization. The respective projection is given in (3.22). In the case of segmentation with moment constraints, the respective constraints can either be fulfilled via the soft constraints or the hard constraints described in Sec. 4.3.

4.4.1 Minimization with Soft Constraints

Soft constraints are implemented by adding additional terms to the cost function, as has been done in (4.6), (4.10) and (4.12). Minimization is then performed by computing the corresponding Euler-Lagrange equations with the additional terms, weighted by the parameters λ_{ki}. Depending on the choice of the λ_{ki} this can lead to solutions that prefer shapes that fulfill the constraints, but do not necessarily exactly fulfill them. For higher order constraints the number of additional parameters $\lambda_{k1}, \dots, \lambda_{kd}$ which have to be optimized increases, with k being the order of the respective moment. This makes an experimental estimation of the values for the λ_{ki} elaborate. An alternative way is to compute optimal values via a gradient ascent procedure.

Area Soft Constraint

For the area constraint the corresponding Euler-Lagrange equation is calculated by derivation of (4.6). The constraint parameter c can be formulated in the integral via division by the size of the image domain $\int_\Omega \mathrm{d}x = |\Omega|$. The resulting equation system is given by:

$$0 = -2\lambda_0 \left(\int_\Omega u - \frac{c}{|\Omega|} \,\mathrm{d}x \right) . \tag{4.18}$$

The area update term for u is equal for all points in Ω because the respective term in (4.18) is independent of a single point x. The constraint parameter c is equally distributed to the update values of all points. The update value is positive if the area of the current segmentation is smaller than the desired area constraint, and negative if it is greater. The update value is zero if the constraint is exactly fulfilled.

The parameter λ_0 can be chosen manually to determine how large the influence of the area constraint should be on the solution. Another option would be to perform a gradient ascent in λ_0. This would be corresponding to a Lagrange multiplier.

Centroid Soft Constraint

Derivation of the centroid soft constraint (4.10) yields the Euler-Lagrange equation

$$0 = -2 \sum_{i=1}^{d} \lambda_{1i} \left(\int_\Omega (\mu_i - x_i) u \,\mathrm{d}x \, (\mu_i - x_i) \right) , \tag{4.19}$$

where μ_i and x_i denote the i-th elements of vectors μ and x. The update term for the centroid constraint is dependent on the location of the point x with respect to the constraint centroid μ. The centroid soft constraint yields d additional terms to the equation system, as well as d additional optimization parameters $\lambda_{11}, \dots, \lambda_{1d}$.

Covariance Soft Constraint

For the covariance constraint the following term arises:

$$0 = -2 \sum_{i=1}^{d} \sum_{j=1}^{d} \lambda_{2ij} \left(\int_{\Omega} \xi_{ij}(x)\, u \,\mathrm{d}x\, \xi_{ij}(x) \right) \tag{4.20}$$

with $\xi_{ij}(x) = a_{ij} - (\mu_i - x_i)(\mu_j - x_j)$. The term contains double entries because $a_{ij} = a_{ji}$ which is due to the symmetry of the covariance matrix. Therefore, instead of d^2 terms, only $d(d+1)/2$ additional terms are needed.

Higher Order Soft Constraints

Similar functions can be derived for the higher order moment constraints. The corresponding Euler-Lagrange term for an arbitrary moment of order k yields the more general formulation:

$$0 = -2 \sum_{i_1=1}^{d} \cdots \sum_{i_k=1}^{d} \lambda_{k i_1 \dots i_k} \left(\int_{\Omega} \xi_{i_1 \dots i_k} u \,\mathrm{d}x\, \xi_{i_1 \dots i_k} \right), \tag{4.21}$$

$$\text{with} \quad \xi_{i_1 \dots i_k} = a_{i_1 \dots i_k} - \prod_{l=1}^{k} (\mu_{i_l} - x_{i_l}) \,. \tag{4.22}$$

In this formulation some terms appear multiple times, similar to the covariance constraint. Because of the symmetry of the constraint tensors, all permutations σ of $i_1 \dots i_k$ have $a_{i_1 \dots i_k} = a_{\sigma(i_1) \dots \sigma(i_k)}$.

4.4.2 Projection to the Moment Constraint Sets

An alternative to the soft constraints are hard constraints where respective constraints are enforced by projection. The moment hard constraints presented in Sec. 4.3 can be implemented by projection onto the constraint sets during the optimization process. The projection method has the advantage over soft constraints, that no additional parameters need to be optimized. In the case of moment hard constraints the constraints can be enforced during the optimization by back-projecting the current segmentation u to the intersection of the respective constraint sets after every iteration. The respective orthogonal projection is the nearest $\hat{u}$ that has a minimum distance to the current u and fulfills the following two types of convex sets

1. the convex set $\mathcal{B}$, i.e. $u \in [0, 1]$ and

2. the intersection of the respective moment constraints.

Because all moment constraints presented in this chapter are linear in u, an orthogonal projection can be directly computed for all combinations of moment constraints. Hence, iterative projections are needed between two sets only – independent of the order or the number of the moment constraints. The following will show how these projections can be computed and implemented. A special case arises in the case of the area constraint, because the projection to the intersection of the area and the range constraint can be computed in a single step. In the other cases the Dykstra projection algorithm [28] is used to iteratively project onto the intersection of the convex sets.

Projection to Area Constraint

The projection onto the range $[0, 1]$ and the area constraint in (4.4) can be combined in one step.

For any $u_0 \in \mathcal{B}$, the projection u onto those constraints has to solve the convex program

$$\begin{aligned} u = \arg\min_{v \in \mathcal{B}} \quad & \frac{1}{2} \left\| v - u_0 \right\|_2^2 \\ s.t. \quad & v(x) - 1 \leq 0 \quad \forall x \\ & -v(x) \leq 0 \quad \forall x \\ & \left\| v \right\|_1 - c = 0. \end{aligned}$$

By means of the Karush-Kuhn-Tucker conditions for convex problems, u is the orthogonal projection if the functions u, $\xi_0, \xi_1 : \Omega \to \mathbb{R}$ and a scalar $\nu \in \mathbb{R}$ fulfill the conditions

$$u(x) - u_0(x) + \xi_1(x) - \xi_0(x) + \nu = 0 \quad \forall x \tag{4.23}$$
$$u(x) \leq 1 \wedge \xi_1(x) \geq 0 \quad \forall x \tag{4.24}$$
$$u(x) \geq 0 \wedge \xi_0(x) \geq 0 \quad \forall x \tag{4.25}$$
$$\xi_1(x) = 0 \vee u(x) = 1 \quad \forall x \tag{4.26}$$
$$\xi_0(x) = 0 \vee u(x) = 0 \quad \forall x \tag{4.27}$$
$$\left\| u \right\|_1 = c. \tag{4.28}$$

With the following method a solution for the conditions (4.23)–(4.28) be found:

1. Initialize $u := u_0, \xi_1 := 0, \xi_0 := 0, \nu := 0$.
2. For all $x \in \Omega$ with $u(x) > 1$, set $u(x) := 1, \xi_1(x) := u_0(x) - 1$.
3. For all $x \in \Omega$ with $u(x) < 0$, set $u(x) := 0, \xi_0(x) := -u_0(x)$.

All constraints except the area constraint (4.28) are now fulfilled.

Without loss of generality, let the projection of u_0 to $[0, 1]$ have a smaller norm than c. If not, set $u_2(x) := 1 - u(x)$ and $c_2 := \|\Omega\| - c$, switch ξ_0 and ξ_1, perform the projection onto the volume c_2 as described in the following, and afterwards again reflect u_2 at 1 and switch ξ_0 and ξ_1.

To fulfill the area constraint (4.28), we can now raise the sum $u(x) + \xi_1(x) - \xi_0(x)$ by the same amount in all pixels, and adjust ν such that equation (4.23) is fulfilled. If $0 < u(x) < 1$, raise $u(x)$, if $u(x) = 1$, raise $\xi_1(x)$, if $\xi_0(x) > 0$, lower $\xi_0(x)$ by the respective values.

Unfortunately, the difference ν is non-trivial, because the area constraint (4.28) depends on u, not on $u + \xi_1 - \xi_0$. Therefore, we have to employ Algorithm 4.4.1: Afterwards, we have to adjust the last iteration, if $\Delta_c < 0$.

Algorithm 4.4.1 Simple Projection Algorithm

```
$\nu \leftarrow 0$
$\Delta_c \leftarrow c - \|u\|$
while $\Delta_c > 0$ do
    $\Delta_u \leftarrow \min\left\{\min_{\{x\in\Omega|\xi_0(x)>0\}}\{\xi_0(x)\}, \min_{\{x\in\Omega|u(x)<1\}}\{1 - u(x)\}\right\}$
    $\Delta_k \leftarrow 0$
    for all $x \in \Omega$ do
        if $\xi_0(x) > 0$ then
            $\xi_0(x) \leftarrow \xi_0(x) - \Delta_u$
        else if $u(x) < 1$ then
            $u(x) \leftarrow u(x) + \Delta_u$
            $\Delta_k \leftarrow \Delta_k + \Delta_u$
        else
            $\xi_1(x) \leftarrow \xi_1(x) + \Delta_u$
        end if
        $\nu \leftarrow \nu + \Delta_u$
    end for
    $\Delta_c \leftarrow \Delta_c - \Delta_k$
end while
```

This algorithm computes the desired projection and it does not require to explicitly store ξ_0 and ξ_1, as the sum in (4.23) can be stored in one field u and the checks in the algorithm above can be performed on u as well.

Unfortunately though, the algorithm requires several $O(n)$ computations in the regression to find the minima and in the update steps.

However, if we sort the discrete pixel positions by their value u_0, and remap them after the projection, the projection can be performed very easily with Algorithm 4.4.2. Let us assume now, that u_0 is stacked to a vector and monotonically decreasing in x from pixel $x = 1$ to pixel $x = |\Omega|$.

Algorithm 4.4.2 Projection Algorithm with Sorting

```
$\nu \leftarrow 0$
$\Delta_c \leftarrow c - \|u_0\|$
$x_1 \leftarrow \min\{x | u_0(x) < 1\}$
$x_2 \leftarrow \min\{x | u_0(x) < 0\}$
**while** $\Delta_c > 0$ **do**
  $\Delta_1 \leftarrow 1 - u_0(x_1) + \nu$
  $\Delta_2 \leftarrow -u_0(x_2) + \nu$
  **if** $\Delta_1 > \Delta_2$ **then**
    $\nu \leftarrow \nu - \min\left\{\Delta_2, \frac{\Delta_c}{x_2 - x_1}\right\}$
    $\Delta_c \leftarrow \Delta_c - \min\{(x_2 - x_1)\Delta_2, \Delta_c\}$
    $x_2 \leftarrow \min\{x | u_0(x) - \nu < 0\}$
  **else**
    $\nu \leftarrow \nu - \min\left\{\Delta_1, \frac{\Delta_c}{x_2 - x_1}\right\}$
    $\Delta_c \leftarrow \Delta_c - \min\{(x_2 - x_1)\Delta_1, \Delta_c\}$
    $x_1 \leftarrow \min\{x | u_0(x) - \nu < 1\}$
  **end if**
**end while**
```

Fig. 4.5 demonstrates the input and output of Algorithm 4.4.2, assuming that u_0 and u are sorted in decreasing order. The optimal update step ν that needs to be added to u_0 in order to fulfill both the area constraint $\int u \, \mathrm{d}x = c$ and the constraint $u \in [0, 1]$, is computed in two steps of Algorithm 4.4.2. Note that the output of Algorithm 4.4.2 is the scalar value ν that is afterwards added to the original u_0 before sorting. Hence the sorting step does not need to be reversed.

After Algorithm 4.4.2 terminates, ν is subtracted from $u_0(x)$ in every pixel to get u, u is clamped to $[0, 1]$ and the difference put into the (virtual) ξ_0 and ξ_1. Now all KKT conditions are fulfilled and u is the desired projection. Note that every assignment in the algorithm is performed at most $2n$ times and the algorithm only requires constant memory. Therefore, its complexity is dominated by the sorting algorithm, for example $\mathcal{O}(n \log n)$.

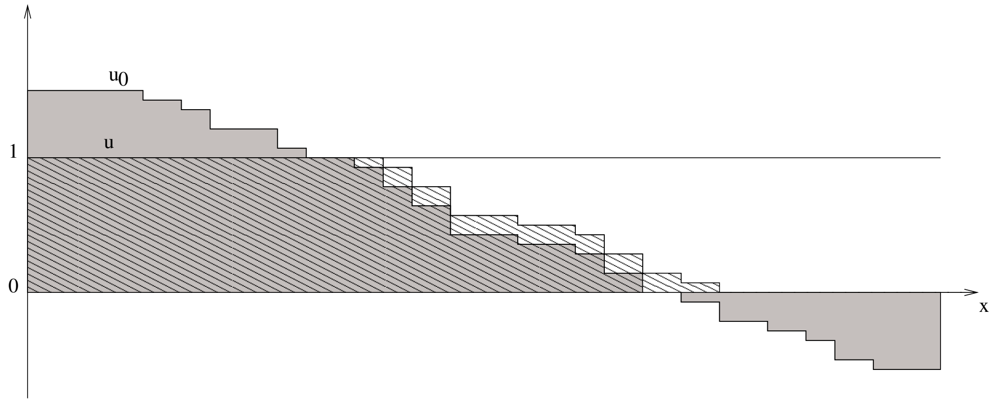


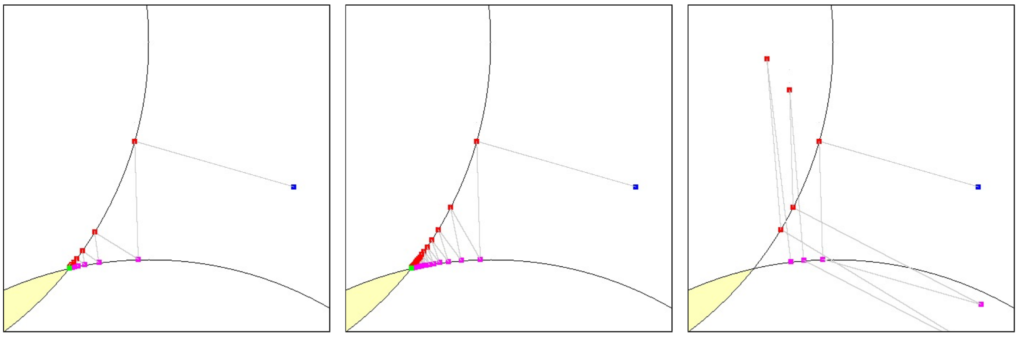
Figure 4.5: Area of u before and after applying Algorithm 4.4.2. The gray area denotes the area before the projection: neither the area constraint $\int u \, \mathrm{dx} = c$ nor the constraint $u \in [0, 1]$ is fulfilled. The shaded area shows the area of u after projection: both constraints are fulfilled after two steps of the algorithm.

Projection to Centroid, Covariance and Higher Order Moment Constraints

The projections to any number of moment constraints can be summarized in one step because the moment constraints give rise to linear sets. Hence, projections are computed between the convex set $[0, 1]$ and the intersection of the moment constraints.

Dykstra's Projection Algorithm For moments of an order higher than 0 the orthogonal projection to the intersection of the set $[0, 1]$ and the respective moment constraints can be computed using the projection algorithm of Dykstra [28]. The algorithm projects to the intersection of convex sets by alternatingly projecting to the respective sets. In this case u is projected to the intersection of the set $[0, 1]$ and the linear moment constraint sets. The solution is found by back-projecting the previous projection for each step. This leads to an algorithm that converges to the solution, although slow compared to a simple iterative projection. Fig. 4.6 shows a comparison of iterative projection and Dykstra's projection algorithm using the example of projecting a point to the intersection of two disks.

Centroid Constraint In the following the projection formula for the centroid constraint will be shown. An equivalent formulation of (4.7) is the



(a) Iterative projection

(b) Dykstra projection

(c) Dykstra back projection steps

Figure 4.6: Comparison of iterative projection (a) and Dykstra's projection algorithm (b). The blue dot is projected to the intersection of two disks, indicated by the yellow area. Intermediate steps are depicted in red. In this example, both algorithms converge to the nearest point in the set, indicated by the green dot while the Dykstra algorithm needs more iterations due to the back projection steps (c).

constraint set

$$\mathcal{C}_1 = \left\{ u \in \mathcal{B} \mid \int_\Omega (\mu_i - x_i)\, u \,\mathrm{d}x = 0, \forall 1 \leq i \leq d \right\}. \tag{4.29}$$

The orthogonal projection $\hat{m}$ that projects u to the nearest $\hat{u}$ in $\mathcal{C}_1$ is then given by $\hat{u} = u + \hat{m}$. Because $\mathcal{C}_1$ is linear in u this projection is unique and

$$\hat{m} = \arg\min_{m \in \mathcal{M}} \|m\| \tag{4.30}$$

$$\text{with} \quad \mathcal{M} = \left\{ m \in \mathcal{B} \mid \int_\Omega (\mu_i - x_i)\, m \,\mathrm{d}x = -\int_\Omega (\mu_i - x_i)\, u \,\mathrm{d}x, \forall 1 \leq i \leq d \right\}. \tag{4.31}$$

The left-hand sides of the constraint equations in $\mathcal{M}$ are linear in m, while the right-hand sides are independent of m.

Then the following projection theorem holds [100]:

Theorem 2. *The orthogonal projection $\hat{m}$ (projection of minimum norm) to the set of constraints $(m, y_i) = c_i, \forall 1 \leq i \leq d$ is given by*

$$\hat{m} = \sum_{i=1}^{d} \beta_i y_i \tag{4.32}$$

where the coefficients $\beta_i \in \mathbb{R}$ are the solution to the linear equation system

$$(y_1, y_i)\beta_1 + \cdots + (y_d, y_i)\beta_d = c_i \tag{4.33}$$

Proof. A proof can be found in [100]. □

For the centroid constraint set the y_i and c_i are given by

$$y_i = \mu_i - x_i,\ c_i = -\int_\Omega (\mu_i - x_i)\, u \,\mathrm{d}x, \quad 1 \le i \le d, \tag{4.34}$$

reducing the problem to solving a linear system of equations of the size $d \times d$ to obtain the coefficients $\beta_1, \dots, \beta_d$. Applying Theorem 2 yields the orthogonal projection

$$\hat{m} = \sum_{i=1}^{d} \beta_i(\mu_i - x_i). \tag{4.35}$$

This means in particular that points that have a high distance to the constraint centroid μ, get assigned a higher value to change. The reason is the orthogonal projection that finds a projection where the smallest change in u is achieved that fulfills the constraint. Points with a high distance to μ contribute higher values to the sum (4.35).

Covariance Constraint Similarly, the orthogonal projection $\hat{m}$ to the covariance constraint can be computed:

$$\hat{m} = \sum_{i=1}^{d}\sum_{j=1}^{d} \beta_{ij} \left(a_{ij} - (\mu_i - x_i)(\mu_j - x_j)\right), \tag{4.36}$$

where a_{ij} are the entries of constraint matrix A. Here we solve a $d^2 \times d^2$ system of linear equations to obtain the coefficients β_{ij}. If we exploit the symmetry of A and merge terms that appear duplicate times in the sum, the number of coeffiences reduces to

$$d' = \sum_{i=1}^{d}\sum_{j=i}^{d} 1 = \frac{1}{2}d(d+1), \tag{4.37}$$

which is still in the same complexity class $O(d^2)$, however reduces the number of coefficients that need to be computed by a factor of almost 2.

Moment Constraints of Arbitrary Order Higher order moment constraints and combinations of different order moment constraints can be computed in an analogous way. The general projection for a moment of order k can be computed by

$$\hat{m} = \sum_{i_1=1}^{d} \cdots \sum_{i_k=1}^{d} \beta_{i_1 \dots i_k} \left(a_{i_1 \dots i_k} - \prod_{l=1}^{k} (\mu_{i_l} - x_{i_l}) \right). \qquad (4.38)$$

The corresponding linear equation system has d^k unknowns $\beta_{i_1 \dots i_k}$.

Orthogonal Projections in a Two-Dimensional Domain

For image segmentation the domain Ω usually is two dimensional, i.e. $\Omega \subset \mathbb{R}^2$ and $d = 2$. The projection onto the centroid constraint $\mathcal{C}_1$ is then given by

$$\hat{m} = \beta_1 \cdot (\mu_1 - x_1) + \beta_2 \cdot (\mu_2 - x_2), \qquad (4.39)$$

and the coeffients $\beta_1, \beta_2 \in \mathbb{R}$ are the solution of the 2×2 linear equation system

$$\begin{aligned} (\mu_1 - x_1)^2 \beta_1 + (\mu_2 - x_2)(\mu_1 - x_1)\beta_2 &= -\int_\Omega (\mu_1 - x_1)\, u \,\mathrm{d}x \\ (\mu_1 - x_1)(\mu_2 - x_2)\beta_1 + (\mu_2 - x_2)^2 \beta_2 &= -\int_\Omega (\mu_2 - x_2)\, u \,\mathrm{d}x. \end{aligned} \qquad (4.40)$$

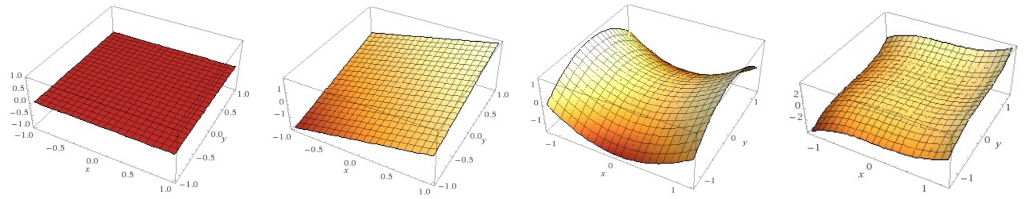
The orthogonal projection to the covariance constraint $\mathcal{C}_2$ is given by

$$\hat{m} = \beta_{11}(a_{11} - (\mu_1 - x_1)^2) + 2\beta_{12}(a_{12} - (\mu_1 - x_1)(\mu_2 - x_2)) + \beta_{22}(a_{22} - (\mu_2 - x_2)^2) \qquad (4.41)$$

and the coeffients $\beta_{11}, \beta_{12}, \beta_{22} \in \mathbb{R}$ are the solution of a 3×3 linear equation system (4.33) with the parameters

$$\begin{aligned} y_1 &= a_{11} - (\mu_1 - x_1)^2 \\ y_2 &= a_{12} - (\mu_1 - x_1)(\mu_2 - x_2) \\ y_3 &= a_{22} - (\mu_2 - x_2)^2. \end{aligned} \qquad (4.42)$$

Fig. 4.7 shows visualizations of the first four moment constraint projections $\hat{m}$ in a two dimensional domain while the projection for the nth order moment constraint yields a nth order polynomial: $\hat{m} = \sum_{i=1}^{n} a_i x^i y^{n-i}$ with coefficients $a_i \in \mathbb{R}$.



(a) Area Projection

(b) Centroid Projection

(c) Covariance Projection

(d) Skewness Projection

Figure 4.7: Update steps for the first four two-dimensional moment constraint projections.

4.4.3 Optimality Bound

Unfortunately, the threshold theorem [34] guaranteeing optimality for the unconstrained binary labeling problem does not generalize to the constrained optimization problems considered here. Nevertheless, we can prove the following optimality bound:

Proposition 7. *Let* $u^* = \arg\min_{u \in \mathcal{C}} E(u)$ *be a minimizer of the relaxed problem and* E_{opt} *the (unknown) minimum of the corresponding binary problem. Then any thresholded version* $\hat{u}$ *of the relaxed solution* u^* *is within a computable bound of the optimum* E_{opt}*.*

Proof. Since E_{opt} lies energetically in between the minimum of the relaxed problem and the energy of the thresholded version, we have:

$$E(\hat{u}) - E_{opt} \leq E(\hat{u}) - E(u^*). \tag{4.43}$$

□

For the experiments on interactive image segmentation that are shown in this chapter this bound was measured on average around 5%. How to assure that the binarized version still exactly fulfills the moment constraints remains an open challenge.

4.5 Ratio Constraints

Variants of the moment constraints give rise to other possibilities to impose shape constraints. In [111] an extension of the area constraint to multi-label image segmentation was presented. Another example is the generalization of the covariance constraint to scale-invariant ratio constraints imposing a constraint on the relation of width to height of a shape. This concept will be presented in the following.

4.5.1 Constraining the Ratio of Shape Dimensions

The covariance of a two dimensional shape can be applied to impose constraints on the ratio of width and height of a shape, yielding a scale-invariant prior for segmentation. We can impose the ratio of a shape to be constrained by a given $\sigma \in \mathbb{R}$ by constraining u to lie in the set

$$\mathcal{R} = \left\{ u \in \mathcal{B} \;\middle|\; \frac{\int_\Omega (x_1 - \mu_1)^2 u \,\mathrm{d}x}{\int_\Omega (x_2 - \mu_2)^2 u \,\mathrm{d}x} = \sigma \right\} \tag{4.44}$$

A linear formulation of the constraint in (4.44) is given by

$$\int_\Omega \left((x_1 - \mu_1)^2 - \sigma (x_2 - \mu_2)^2 \right) u \,\mathrm{d}x = 0 \tag{4.45}$$

The set $\mathcal{R}$ is also convex, which can be seen in analogy to Proposition 3. Fig. 4.8 shows an example for image segmentation with ratio constraint.

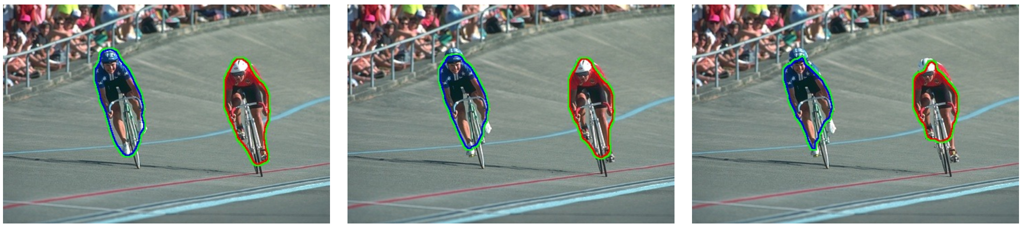


Figure 4.8: Image segmentation with different values for the ratio constraint parameter σ.

4.5.2 Projection to Ratio Constraints

The projection of a segmentation u to the ratio constraint set $\mathcal{R}$ yields:

$$\hat{m} = \beta \left((x_1 - \mu_1)^2 - \sigma (x_2 - \mu_2)^2 \right), \tag{4.46}$$

$$\text{with} \quad \beta = -\frac{\int_\Omega \alpha \, u \,\mathrm{d}x}{\int_\Omega \alpha^2 \,\mathrm{d}x} \quad \text{and} \quad \alpha = (x_1 - \mu_1)^2 - \sigma (x_2 - \mu_2)^2. \tag{4.47}$$

The projection derives from Theorem 2.

4.6 Interactive Image Segmentation with Area, Centroid and Covariance Constraints

For the application of interactive image segmentation the experiments are limited to moments up to 2nd order, i.e. area, centroid and covariance, because they can be intuitively transmitted via mouse interaction.

In this section a qualitative and quantitative evaluation of the proposed method on medical imagery and other real-world images is presented. For all experiments the edge weight function was set to $g(x) = 1$ and the data term to $f(x) = \log(p_{bg}(I(x))) - \log(p_{obj}(I(x)))$ for an input image $I : \Omega \to \mathbb{R}$ for gray value images and $I : \Omega \to \mathbb{R}^3$ for color images. The likelihoods p_{obj} and p_{bg} are computed using color or gray-scale histograms, respectively, from inside and outside regions defined by the user input. Respective moment constraints on centroid, area or covariance structure are imposed by mouse interactions. In all experiments shown moment constraints are enforced by iteratively projecting solutions to the respective constraint sets after each iteration of the optimization process. Typical run-times on the GPU are around 1 second for an image of the size 300 × 400.

4.6.1 Shape Priors from Ellipses

The constraint parameters for the moment contraints can be imposed in different ways. One way to impose the parameters of the lower order moments area, centroid and covariance is to transmit them via a two-click mouse interaction in an intuitive way: by drawing an ellipse around the object of interest. The area and centroid constraint can be directly computed as the ellipse's area and center point. The covariance constraint entries correspond to the ellipse's radius and axis orientations. In particular, the eigenvalues $\sigma_1, \sigma_2 \in \mathbb{R}$ of the covariance constraint matrix $A \in \mathbb{R}^{2\times 2}$ can be computed from the major and minor radius $r_1, r_2 \in \mathbb{R}$ of the ellipse with

$$\sigma_i = \frac{1}{2} r_i^2, \quad i = 1, 2. \tag{4.48}$$

The covariance constraint is then given by its eigendecomposition

$$A = V \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} V^{-1}, \tag{4.49}$$

where the matrix $V \in \mathbb{R}^{2\times 2}$ is a rotation according to the respective orientation of the semi-axes of the ellipse.

4.6.2 Image Segmentation Results

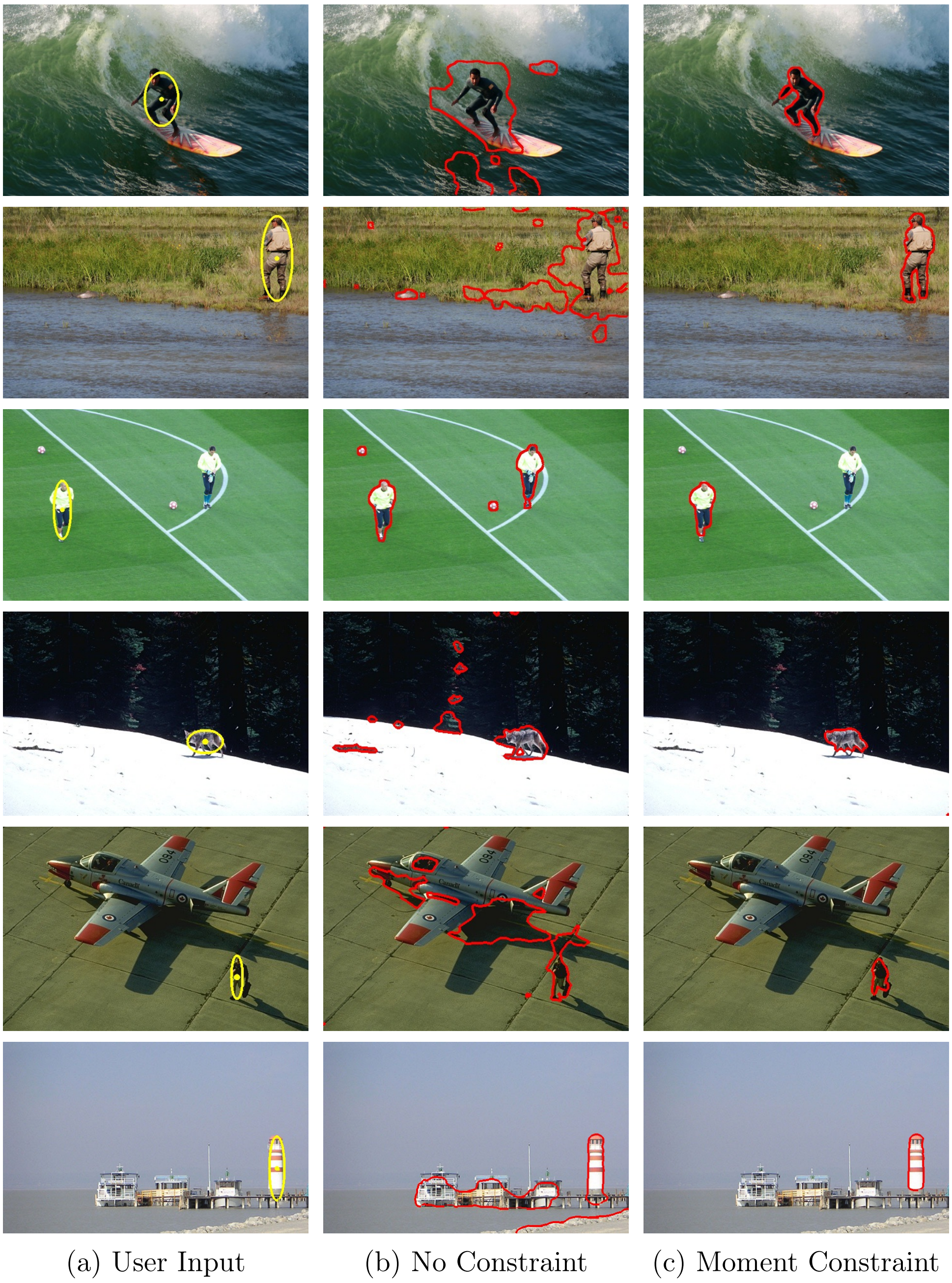
Fig. 4.9 shows how moment constraints can improve segmentation of real-world images. The data term is based on RGB histograms. The purely color-based segmentations without moment constraints shown in the second column demonstrate that the color distributions of respective objects are not sufficiently different to discriminate the objects of interest. The third column of the figure shows the segmentation results with constraints on area, centroid and covariance. All moment constraints are extracted from the user-specified ellipse, allowing a deviation of 10% for each constraint to handle imprecise user input. The moment constraints allow to quickly disambiguate the color information leading to substantial improvements of the segmentation.

4.6.3 Comparison to Scribble Based Segmentation

Many interactive image segmentation methods that use manual input from the user implement scribbles to obtain initial values for the histograms that generate the data term f. Scribbles in this context are selected pixels marked by the user as foreground and background. Segmentations are be implemented in discrete settings [123], or continuous settings [141]. In this section a comparison to a segmentation method similar to [141] is presented.

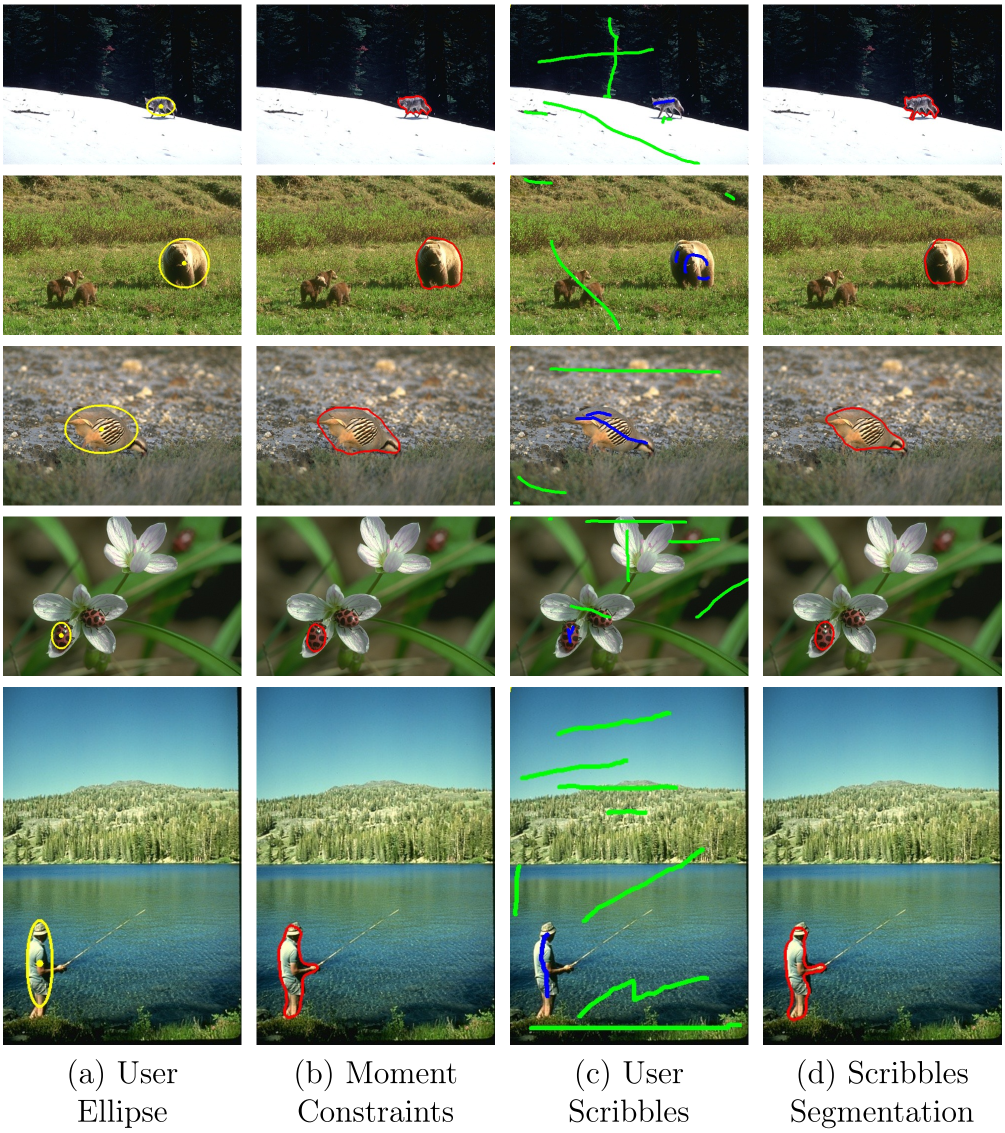
Fig. 4.10 shows a comparison of two segmentation methods with priors on color and location from user input. The first two columns show the presented method with moment constraints and the third and fourth columns a segmentation method based on user scribbles. The purpose of this experiment was to compare the effort that needs to be brought in by the user in order to sufficiently segment the respective object of interest from the background. For the scribble segmentation the user draws mouse strokes in two different colors: blue for foreground and green for background. Histograms for the data term f are computed from these selected pixels. Segmentation results with user scribbles were computed by minimizing functional (4.3), using the same method and same parameters as in the case of the moment constraints, except that the moment constraints were replaced by local constraints, which means that all pixels that were marked by the user as foreground belong to the segmented region, and the background pixels are not contained by the segmented region.

The figure shows that segmentations with user scribbles need significantly more mouse interaction compared to segmentation with moment constraints. This implies that moment constraints are able to substantially simplify the task of image segmentation for the user.



(a) User Input (b) No Constraint (c) Moment Constraint

Figure 4.9: Image segmentation without and with moment constraints. (a): The ellipse is placed at the approximate size and location of the object. (b): Segmentation results without constraints using only the histograms. (c): Segmentation results with constraints on area, centroid and covariance.


(a) User Ellipse

(b) Moment Constraints

(c) User Scribbles

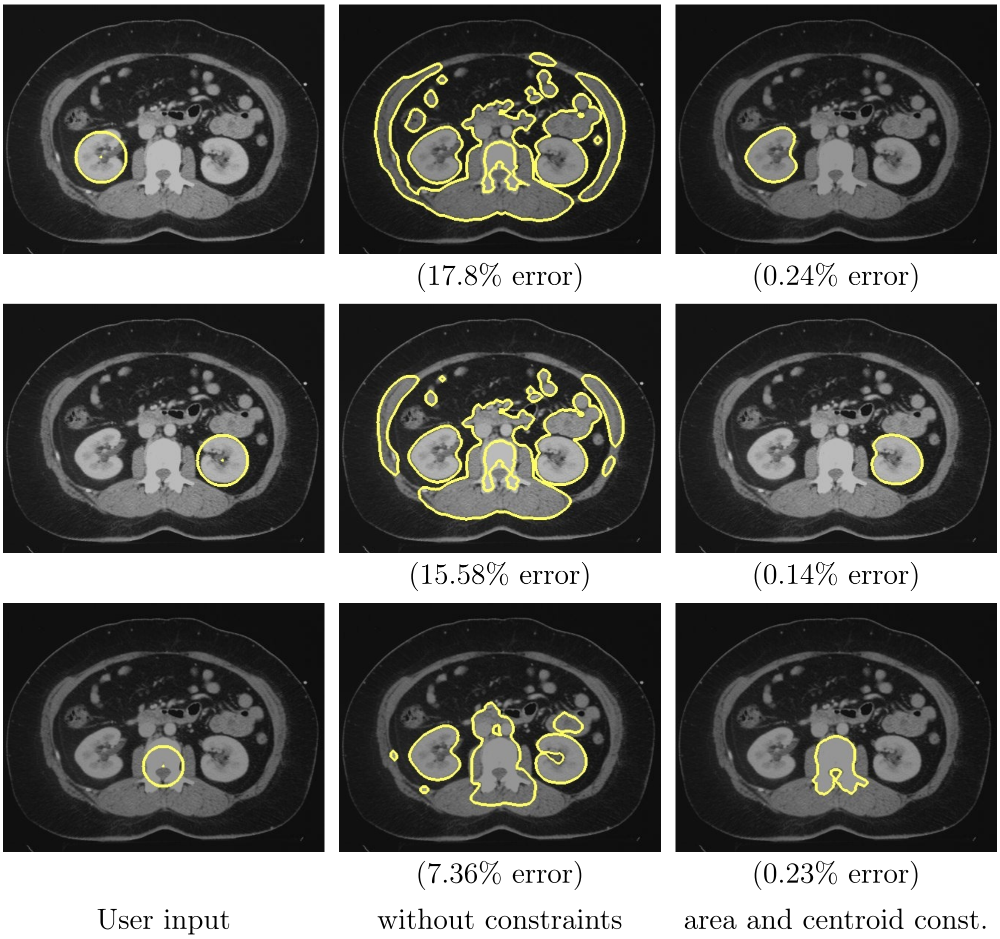
(d) Scribbles Segmentation

Figure 4.10: Comparison of segmentation with moment constraints and user scribbles: Segmentation with user scribbles needs more mouse interaction to obtain similar results, since the implementation with moment constraints needs just two mouse clicks for drawing the respective ellipse.

4.6.4 Quantitative Evaluation on Medical Images

This section presents an evaluation of segmentations with and without moment constraints on a set of medical images, and a quantitative comparison of the results to manually labelled ground truths.

Area and Centroid Constraints



(17.8% error) (0.24% error)

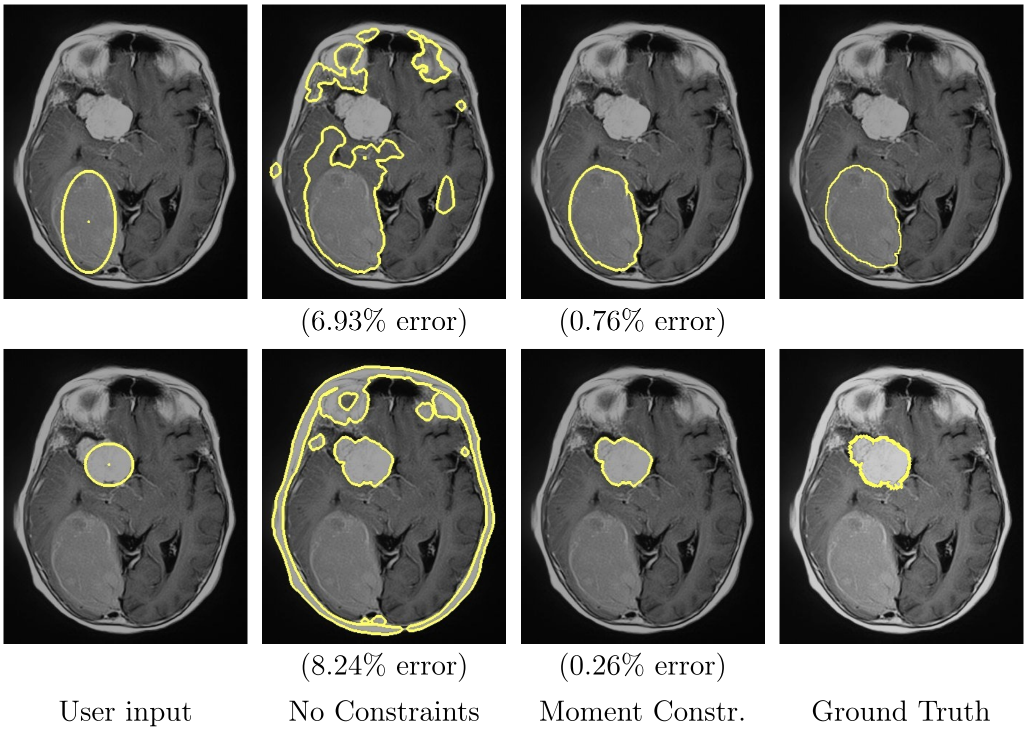
(15.58% error) (0.14% error)

(7.36% error) (0.23% error)

User input without constraints area and centroid const.

Figure 4.11: Segmentation of a CT image with kidneys and spine. The centroid and area constraints enable to specify the approximate location and size of the desired object that should be segmented. Imposing these moment constraints during optimization leads to drastic improvements in the segmentation.

Fig. 4.11 shows a comparison of segmentation with and without a constraint on the area and centroid for a CT image of kidneys and spine: without constraints no shape information is taken into account for the segmentation, resulting in a segmentation that includes many different regions. Enabling the area and centroid constraints leads to segmentations that prefer the center and the size of the circle that was clicked by the user. This leads to substantial improvements of the segmentations without affecting the fine-scale boundary estimation.



(6.93% error) (0.76% error)

(8.24% error) (0.26% error)

User input No Constraints Moment Constr. Ground Truth

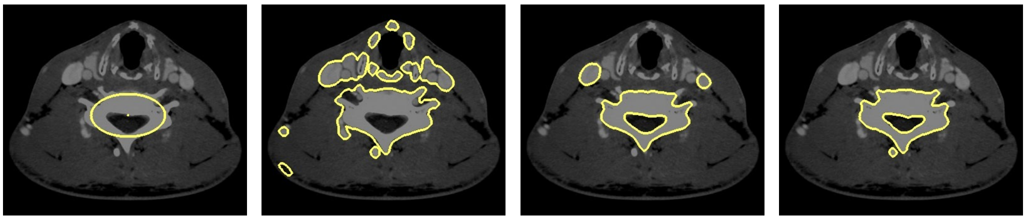
Figure 4.12: Tumor extraction in brain MR images using segmentation with and without constraints on covariance and area. While the algorithm does not require local boundary information, constraining its second order moments by a simple user interaction suffices to generate the desired segmentation.

Including Covariance Constraints

More sophisticated structures can be specified when including second order moments. Since covariance matrices can be represented by ellipsoids, an intuitive user input is achieved by clicking an ellipse with the mouse. The axes of the ellipse define the entries of the corresponding covariance matrix, while the center and area of the ellipse define the centroid and area constraints. Fig. 4.12 and 4.13 show segmentations with and without constraints resulting from user defined ellipses describing the approximated size, location and shape of the desired object.

Quantitative Performance Evaluation

The previous experiments have shown that the user-specified moment constraints allow to visibly improve the segmentation. To quantify this improvement, Table 4.1 shows average relative errors (i.e. the percentage of



User input

(6.14% error) without constraints

(1.41% error) with area constraint

(1.04% error) covariance and area constraint

Figure 4.13: Segmentation without and with constraints for a CT image of the neck. Constraining the area yields a segmentation which prefers the size of the ellipse that was clicked by the user, resulting in less incorrectly labeled pixels, compared to the segmentation without constraints. The covariance constraint additionally considers the dimensions of the ellipse yielding an even more accurate segmentation.

incorrectly labeled pixels per image) with standard deviations for an evaluation of the segmentation without constraint, with area constraint only, and with area, centroid and covariance constraint, respectively. Some of the images that were used for the tests and their segmentations are shown in Fig. 4.11, 4.12 and 4.13. The table shows that the use of these rather simple and easy to transmit constraints yield a reduction of incorrectly classified pixels by a factor of about 10.

4.6.5 Performance Evaluation of Constraint Projections

Fig. 4.14 shows run times in seconds for projections onto the presented moment hard constraints. It shows a measurement of the number of seconds

	Average relative error
Segmentation without constraint	12.02 % ± 0.89%
with 0th order moment constraint	2.36 % ± 0.11%
with 0th–1st order moment constraint	0.41 % ± 0.05%
with 0th–2nd order moment constraint	0.35 % ± 0.09%

Table 4.1: Average relative errors with standard deviations for segmentation without and with moment constraints.

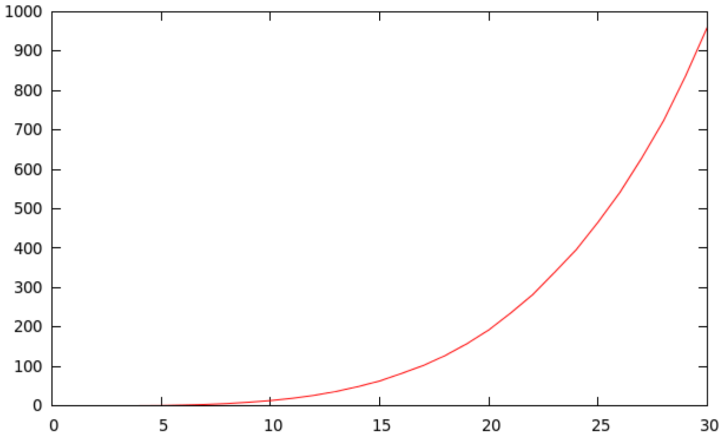


Figure 4.14: Run time (in seconds) vs. number of moment constraints k in projection. The plot shows the number of seconds needed to compute a projection onto the moments of order 0 to k. While lower order moment constraints are computed in a few milliseconds, computation times grow exponentially for increasing numbers of constraints. Fortunately, for interactive segmentation applications one is mostly interested in constraining the low-order moments only, leaving the fine-scale details to be determined by the image data.

that were needed on a 3.4 GHz Intel Core i7-2600 CPU to compute one projection onto all moment constraints of order 0 to k for $k \in \{0, \ldots, 30\}$ using the projection formula (4.38) for projection onto moment constraint sets of arbitrary order. The constraint parameters were obtained by computing the moments of the manually segmented reference shape shown in the upper row of Fig. 4.4 (a). While projections onto the lower order moment constraints are computed in just a few milliseconds, the figure shows the exponentially growing run time for an increasing number of constraints.

The experiment implies that segmentation with moment constraints is applicable to real-time tasks for the lower order moments, whereas higher order moments that can theoretically constrain arbitrary shapes, are applicable merely to off-line segmentations.

4.6.6 Optimality Bounds

The threshold theorem [34] states that in the unconstrained case the thresholded version is a globally optimal solution of the binary energy (4.3). However this is not the case when additional moment constraints are imposed on the segmentation. This sections analyses the difference of the continuous result to the thresholded version. For a continuous solution u^* and a thresholded solution $\hat{u}$ the difference was computed as the relative energy bound

$$e = \frac{E(\hat{u}) - E(u^*)}{E(u^*)}. \tag{4.50}$$

Fig. 4.15 shows segmentation results for u before thresholding, computed with constraints on area, centroid and covariance. The values below the images refers to the energy bound (4.50). In all four experiments that are shown in the figure the solutions converge to binary values and the distance to the thresholded version is below 1%. This implies that segmentations with moment constraints are robust to the chosen threshold. In the experiments, we furthermore observed that the more constraints are imposed on a segmentation, the larger the distance to the thresholded version increases.

4.7 Moment Constraints for Object Tracking

Object tracking over a sequence of images is the problem of finding the shape of an object that was given in the first frame in the subsequent frames of the sequence. This section shows how segmentation with moment constraints can be generalized with a few modifications to a method for object tracking. Constraining the moments of a shape during a sequence of images leads – in combination with the presented method for image segmentation – to a method for object tracking. Given the moments of the shape in the first frame, these moments can be constrained for all subsequent frames as well. u is now a function defined on the image plane Ω evolving over time $T \subseteq \mathbb{R}^+$, i.e. $u : \Omega \times T \to [0, 1]$.

4.7.1 Permanency Constraints

The area constraint can be used to track an object over time: Assuming that the area of the shape does not change over time, the area of a shape is constrained at time t to being equal to the area c_{t-1} of the shape in the previous frame $t - 1$:

$$\min_{u \in \mathcal{B}} E(u) \quad \text{s.t.} \quad \left| \int_\Omega u(x,t) \,\mathrm{d}x - c_{t-1} \right| = 0, \quad \forall t \in T. \tag{4.51}$$

Here c_0 is defined as the area of the ellipse drawn by the user in the first frame.

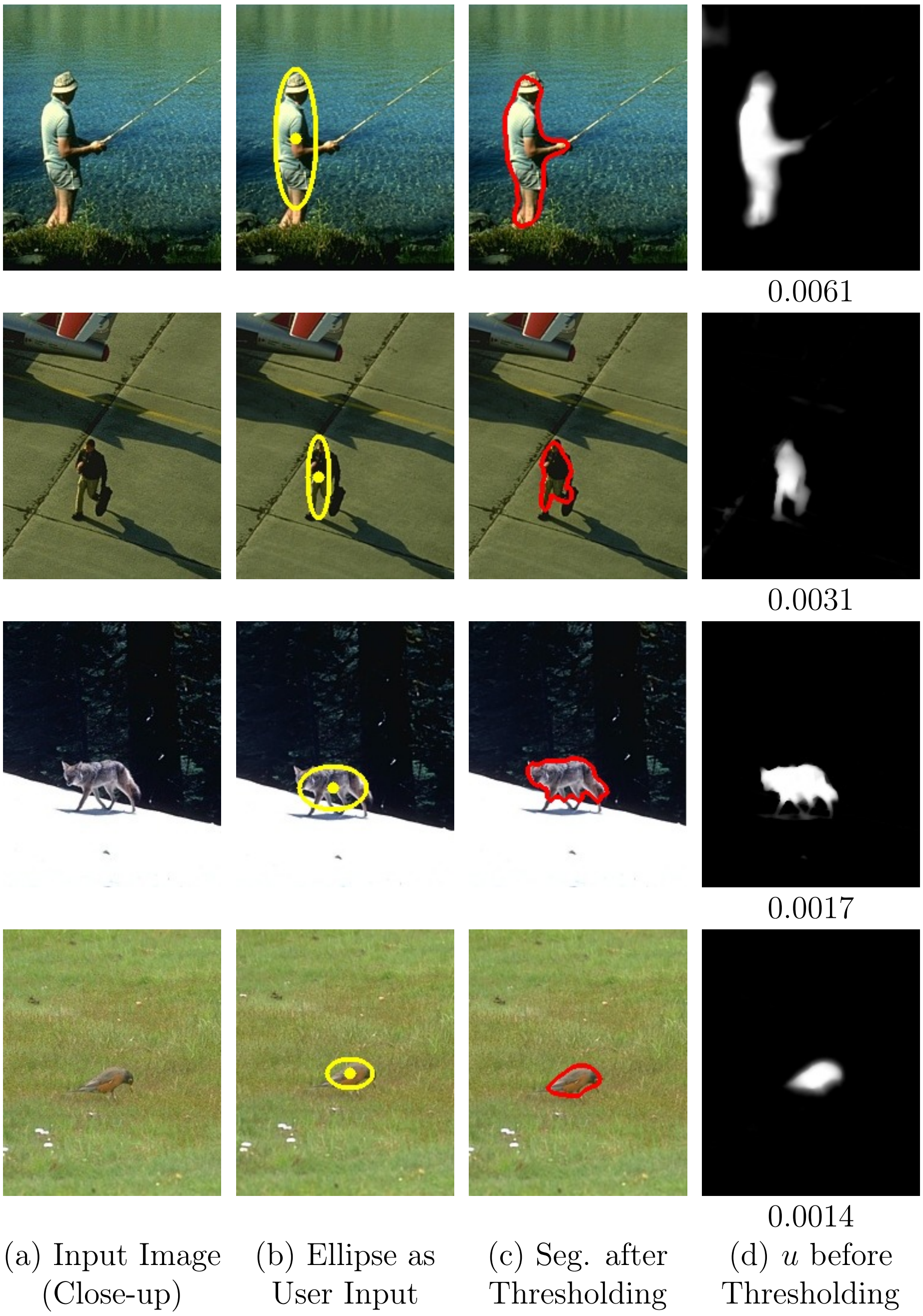

0.0061
0.0031
0.0017
0.0014
(a) Input Image (Close-up)
(b) Ellipse as User Input
(c) Seg. after Thresholding
(d) u before Thresholding

Figure 4.15: Segmentation with the convex moment constraints converges to nearly binary solutions, making the method robust to the chosen threshold. The values below the images in (d) refers to the distance between the continuous result and its corresponding thresholded version.

Alternatively, a small change of the area constraint can be allowed by replacing the $= 0$ in equation (4.51) by $\leq v$ with a constant value $v \in \mathbb{R}$ This enables to model motion of the object towards or away from the camera, or motion of the camera towards/away from the object. Here, v corresponds to the the amount that the area of the shape is allowed to change from one frame to the next, i.e. depends on the velocity of the motion.

4.7.2 Velocity Constraints

Similarly, a constraint can be imposed that the object should not move too far from one frame to the other, which means that the centroid of the shape does not change more than a given value v:

$$\min_{u \in \mathcal{B}} E(u) \quad \text{s.t.} \quad \left| \frac{\int_\Omega x u(x,t)\,\mathrm{d}x}{\int_\Omega u(x,t)\,\mathrm{d}x} - \mu_{t-1} \right| \leq v, \quad \forall t \in T. \tag{4.52}$$

v corresponds to the maximum length of the vector between the centroid in one frame and the centroid in the next frame, and μ_{t-1} is the centroid of u in the previous frame. Again, we define μ_{-1} as the centroid of the ellipse drawn by the user in the first frame in order to obtain an initialization for $t = 0$.

4.7.3 Rotational Constraints

Similar constraints can be imposed on the covariance and higher order moment constraints. For the covariance constraint, for example, rotation of the object can be constrained by assuming that the covariance matrix should not change more than a given value.

Optimization is performed with the same method as explained in Sec. 4.4 while the respective moment constraints are updated in each frame.

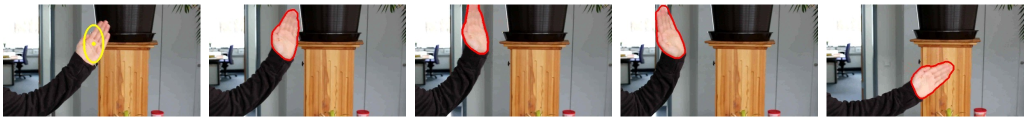


Figure 4.16: Moment constraints for object tracking. Tracking is initialized with an ellipse in the first frame, the moments of which constrain the segmentation in subsequent images. A small deviation of the centroid is allowed to track the moving object. Note that this approach is generic, as no reference shapes have to be previously learned.

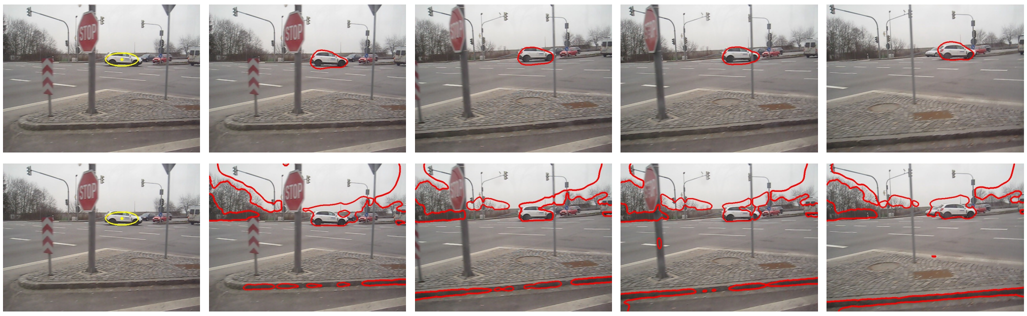


Figure 4.17: Moment constraints for object tracking. Tracking is initialized in form of an ellipse in the first frame (left), from which histograms and constraint parameters are derived. The first row shows results with moment constraints, while a deviation of the centroid is allowed from each frame to the next one to account for the object's motion. The second row shows results of a histogram based tracking without constraints. This comparison shows that moment constraints can realize acceptable real-world object tracking with no previous learning of reference shapes.

4.7.4 Experiments

Fig. 4.16 and 4.17 show how the proposed method can be applied to tracking objects in videos. As can be seen in Fig. 4.17, the purely color-based segmentation does not suffice to correctly segment object from background in the case of non-unique color distributions.

We impose shape information by constraining the low order moments (area, centroid and covariance) throughout the entire image sequence. As can be seen in the first image of each sequence, tracking is initialized with an ellipse of the approximate size and location of the object which is drawn on the first frame of the sequence. This is sufficient user input, since histograms and moment constraint parameters are derived from the ellipse: again, histograms for foreground and background are computed from the inside and outside of the ellipse, respectively, and the constraint parameters for area, centroid and covariance are derived from the ellipse's area, center point and principal axes. The subsequent frames of the video use the histograms and moment constraints from the first frame, allowing a small deviation of the centroid from each frame to the next, which corresponds to a constraint on the maximum velocity. Since no previous learning of shapes is necessary, the approach naturally applies to arbitrary object shapes.

4.8 Conclusion

In this chapter, moment constraints in a convex shape optimization framework were presented. In particular, it was shown that for an entire family of constraints on the area, the centroid, the covariance structure of the shape and respective higher-order moments, the feasible constraint sets are all convex. While global optimality of the resulting segmentations cannot be guaranteed, the computed solutions are independent of initializations and within a known bound of the optimum.

Both qualitative and quantitative experiments on interactive image segmentation using medical and real-world images demonstrated that respective moment constraints can be easily imposed by the user. The application of moment constraints lead to significant improvements of the segmentation results, reducing the average segmentation error from 12% to 0.35%. In contrast to existing works on shape priors in segmentation the use of low-order moment constraints does not require shape learning and is easily applied to arbitrary shapes since the recovery of fine scale shape details is not affected through the moment constraints. Efficient GPU-accelerated PDE solvers allow for computation times of about one second for images of size 300 × 400, making this a practical tool for interactive image segmentation.

In the last section the applicability of lower order moment constraints to object tracking in image sequences was shown. The following chapter describes an extension of the presented approach to scale-aware object tracking in 3D space using RGB-D images.

Chapter 5

Scale-Aware Object Tracking in RGB-D Sequences

This chapter shows how the moment constraints presented in Chapter 4 can be extended to the 3D space, which allows for scale-invariant object tracking in RGB-D. It presents a novel technique for the segmentation of RGB-D images using convex function optimization. The minimization of the proposed function finds optimal segmentations by considering both the color and the depth information. The objective function is extended by moment constraints, which allow to include prior knowledge on the 3D center, surface area or volume of the object in an elementary way. As will be shown in this chapter, the relaxed optimization problem is convex, and thus can be minimized in a globally optimal way leading to high-quality solutions independent of initializations. The approach is validated experimentally on five different datasets. Experiments show that using both color and depth substantially improves segmentations compared to color or depth segmentations only. Furthermore, 3D moment constraints significantly robustify segmentations which proves in particular useful for object tracking.

Parts of this chapter have been published in [7].

5.1 Introduction

Image segmentation and tracking are of central importance in image analysis. Many successful approaches to image segmentation from monochrome or color images have been proposed in the past [22, 141]. Unfortunately, in many real-world applications object and background share similar colors such that purely 2D color-based segmentation methods invariably fail.

With the rise of novel RGB-D cameras like the Microsoft Kinect, inex-

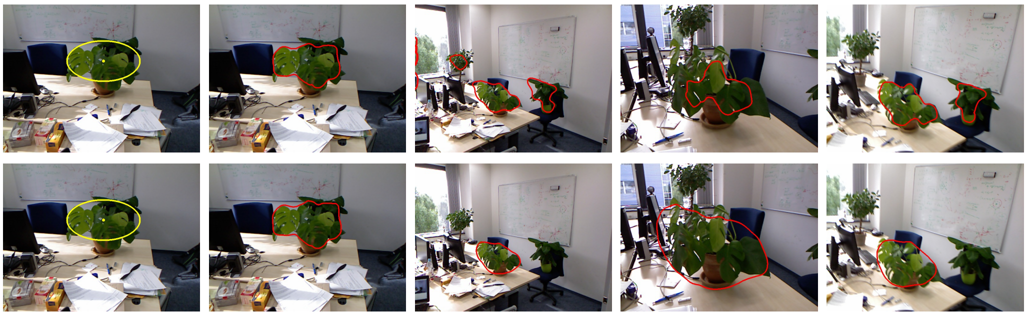


Figure 5.1: Tracking with area constraints: RGB area constraints (first row) are not capable to handle camera motion, whereas the RGB-D area constraints (second row) are scale-invariant.

pensive sensors became available that provide both color images and depth maps synchronized and at high resolution. While depth alone is usually not sufficient to achieve good segmentation results (different objects may share the same depth), it is well-known that the combination of depth and color information outperforms purely color-based segmentation [61] and allows for significant speed-ups of the segmentation process [136]. Moreover, it will be shown in this chapter, that when prior knowledge about the object is available – like for example, its surface area, centroid, or shape covariance matrix – this knowledge can be exploited during object segmentation.

This chapter shows how the convex framework for color image segmentation introduced in Chapter 4 can be extended to RGB-D image data. It contains two main contributions:

- The first contribution is the extension of the data term of respective segmentation energies to incorporate local depth information. As a consequence, the respective algorithm favors a separation of object and background based on both color and depth information: It is therefore able to distinguish structures of the same color but with different depths. As a consequence objects that would be difficult to separate by color or depth alone can be segmented.

- The second contribution is the generalization of the moment constraints introduced in Chapter 4 to scale-awareness using the information of the object's distance from the camera. More specifically, the depth maps enable to impose constraints on the object's *absolute* shape in 3D, whereas purely color based tracking methods can only impose constraints on the object's *projected* shape in the 2D image plane. These constraints can either be specified manually by user input, or automatically extracted from an initial segmentation for example for object tracking. Experiments show that the approach allows to reliably segment and track humans and plants in RGB-D images. Further, it is shown that respective moment constraints can be generalized to the RGB-D setting thereby assuring that – for example – the surface area in 3D space is preserved. In tracking experiments beyond constraining the object's sideways motion we can thus also constrain the motion of the object along the camera axis.

The outline of this chapter is as follows: Sec. 5.2 will give a brief overview of related work on shape priors for RGB-D image segmentation and object tracking. In Sec. 5.3 a scale-aware object tracking method based on convex 3D moment constraints is presented. Sec. 5.4 will show experimental results for object tracking and image segmentation in RGB-D. Sec. 5.5 will conclude with a summary of the main results.

5.2 Related Work

Image segmentation is among the most studied problems in image analysis. Popular algorithms to solve the arising shape optimization problems include level set methods [113], graph cuts [67] or convex relaxation [34], with respective extensions to the multi-region case [37, 27, 148, 94, 32].

Segmentations can be improved using both color and depth information [109] obtained by RGB-D cameras like the kinect.

While it was shown that segmentation results can be substantially improved by imposing shape priors [68, 39, 52], existing approaches have several limitations: Firstly, apart from a few exceptions such as [143, 128], computable solutions are only *locally* optimal thus requiring appropriate initializations and leading to often suboptimal solutions. Secondly, many shape priors require an entire training set of familiar shapes [39, 48], making them impractical for generic interactive image segmentation where the user may have a good idea of what he/she wants but will be hard pressed to construct an entire training set of shapes.

As a remedy it was recently proposed [3] to interactively impose constraints on the lower-order moments of the shape in a convex relaxation framework for image segmentation. The aim of this paper is to generalize these concepts to the problem of RGB-D image segmentation.

5.3 Tracking in RGB-D Sequences with Scale-Aware Shape Constraints

This section will show how moment constraints can be incorporated for RGB-D image segmentation and applied to object tracking in RGB-D sequences.

5.3.1 RGB-D Image Segmentation with Convex Relaxation

With additional depth information $d : \Omega \to \mathbb{R}$ from RGB-D images, the boundary length can be measured in absolute values instead of the image domain. Functional (4.3) can be generalized to

$$E(u) = \int_\Omega f(x)\, u(x)\, dx \, + \int_\Omega d(x)|Du(x)|. \tag{5.1}$$

This formulation compensates the fact that objects that are far away to the camera appear smaller in the image due to perspective projection. Weighting with $d(x)$ allows regularization on the absolute size of the boundary – in contrast to assuming a uniform pixel size as in (4.3).

5.3.2 Moment Constraints for RGB-D Images

In the following, the moments of the segmentation will be successively constrained with depth information. It will be shown how these constraints give rise to nested convex sets. Again, the set $\mathcal{B} = BV(\Omega; [0, 1])$ denotes the convex hull of the set of binary indicator functions $u \in BV(\Omega; \{0, 1\})$ of bounded variation on the domain $\Omega \subset \mathbb{R}^d$.

Area Constraint

The 0-th order moment corresponds to the area of the shape u. In RGB-D the 3D surface area can be computed by

$$\text{Area}(u) := \int_\Omega d^2(x) u(x)\, \mathrm{d}x, \tag{5.2}$$

where $d(x)$ gives the depth of pixel x. Here, we assume that $d(x) = K\tilde{d}(x)$, with K being the focal length of the camera and $\tilde{d}(x)$ being the depth of the pixel measured in meters. Note that $d^2(x)$ corresponds to the size of a back-projected pixel in 3D space, and thus the integral measures the absolute surface area (scaled by K^2) instead of the projected area in the image. This is in contrast to Sec. 4.3.1, where all pixels are treated equally.

The absolute area of the shape u can be imposed to be bounded by constants $c_1 \leq c_2$ by constraining u to lie in the set:

$$\mathcal{C}_0' = \left\{ u \in \mathcal{B} \,\middle|\, c_1 \leq \text{Area}(u) \leq c_2 \right\}. \tag{5.3}$$

The set $\mathcal{C}_0'$ is linearly dependent on u and therefore convex for any constants $c_2 \geq c_1 \geq 0$.

In practice, the area constraint can be imposed exactly by setting $c_1 = c_2$, or it can be imposed by upper and lower bounds on the area. Alternatively, a soft area constraint can be formulated by enhancing the functional (4.3) as follows:

$$E_0'(u, \lambda_0) = E(u) + \lambda_0 \left(\int d^2 u \, \mathrm{d}x - c \right)^2, \tag{5.4}$$

which imposes a soft constraint with a weight $\lambda_0 > 0$ favoring the area of the estimated shape to be near $c \geq 0$. Note that the functional (5.4) is also convex.

Centroid Constraint

The first order moment corresponds to the center of gravity (or *centroid*) of the shape. In RGB-D it can be computed by integrating over all 3D positions of the visible part of the shape, i.e.,

$$\mu(u) := \begin{pmatrix} \overline{x} \\ \overline{d} \end{pmatrix} = \frac{\int_\Omega \binom{x}{d} u \, \mathrm{d}x}{\int_\Omega d^2 u \, \mathrm{d}x}, \tag{5.5}$$

where $\overline{x} \in \mathbb{R}^2$ is the centroid in pixel coordinates and $\overline{d} \in \mathbb{R}$ is the centroid in depth. Together, $\mu \in \mathbb{R}^3$ corresponds to the centroid of the shape in 3D.

Now bounds on the centroid for the object we want to segment can be imposed by constraining the solution u to the set $\mathcal{C}_1'$:

$$\mathcal{C}_1' = \left\{ u \in \mathcal{B} \,\middle|\, \mu_1 \leq \mu(u) \leq \mu_2 \right\}, \tag{5.6}$$

where all inequalities are to be taken point-wise and $\mu_1, \mu_2 \in \mathbb{R}^3$. This imposes the centroid to lie between the two constants $\mu_1 \leq \mu_2$. In particular, for $\mu_1 = \mu_2$, the centroid is fixed.

Proposition 8. *For any constants $\mu_2 \geq \mu_1 \geq 0$, the set $\mathcal{C}_1'$ is convex.*

The proof is analogous to proof 2 in Chapter 4.

Alternatively, the centroid constraint can be formulated as a soft constraint by minimizing the energy:

$$E_1'(u, \lambda_1) = E(u) + \lambda_1 \left| \int_\Omega \left(\mu d^2 - \left(\begin{smallmatrix} x \\ d \end{smallmatrix} \right) \right) u \, \mathrm{dx} \right|^2 , \tag{5.7}$$

which is also convex in u.

Covariance Constraint

The proposed concept can be generalized to central moments of second order. The 3D covariance of a shape u with respect to a specified centroid μ in RGB-D is given by

$$\mathrm{Cov}(u) := \frac{\int_\Omega \left(\binom{x}{d} - \mu \right) \left(\binom{x}{d} - \mu \right)^\top u \, \mathrm{d}x}{\int_\Omega d^2 u \, \mathrm{d}x}. \tag{5.8}$$

The covariance structure can be considered by the following convex set:

$$\mathcal{C}_2' = \left\{ u \in \mathcal{B} \;\middle|\; A_1 \leq \mathrm{Cov}(u) \leq A_2 \right\} \tag{5.9}$$

where the inequality constraint should be taken element wise. Here $\mu \in \mathbb{R}^3$ denotes the centroid and $A_1, A_2 \in \mathbb{R}^{3\times 3}$ denote symmetric matrices such that $A_1 \leq A_2$ element wise. This constraint is particularly meaningful if one additionally constrains the centroid to be μ, i.e. considers the intersection of the set (5.9) with a set of the form (5.6).

Optimization with Moment Constraints

Shape optimization and image segmentation with respective moment constraints can now be done by minimizing convex energies under respective convex constraints. The optimization was implemented using the projection approach as described in Sec 4.4.

5.3.3 Object Tracking with 3D Constraints

The 3D moments of a shape can be used for tracking objects in a sequence of images. Given the moments of the shape in the first frame, constraints can be imposed on segmentations in all subsequent frames. Here, the moments of a shape are computed directly in the 3D space, not in the projection to the image plane. This makes the method independent of the projected size of the object in the image. Without the need of defining a window in which subsequent shapes should be found, the proposed method simply applies the moment constraints of the current frame to the subsequent. The centroid is allowed to change inside a small range to handle motion of the camera and/or the object. The area and covariance are supposed to stay constant in the 3D space over all time frames.

5.3.4 Segmentation Priors from User Input

The data term used throughout the experiments has the form

$$f(x) = \log \frac{p_{\text{bg}}(I(x))}{p_{\text{obj}}(I(x))}. \tag{5.10}$$

Here, $I : \Omega \to \mathbb{R}^n$ refers to an image with n channels. For example, $n = 1$ for depth or gray-scale images, $n = 3$ for color images and $n = 4$ for RGB-D images. The data priors p_{obj} and p_{bg} assign probabilities to each pixel belonging to the object or the background, respectively, and satisfy $p_{\text{obj}} + p_{\text{bg}} = 1$. They were computed from histograms for foreground and background. The moment constraints that are considered in the experiments include the centroid, area and covariance of the shapes.

Both the data prior as well as the moment constraints can be specified by the user. The user can use an intuitive interface to mark the object of interest with an ellipse (see Fig. 5.1). From the pixels within and outside the ellipse, the n-dimensional color/depth/RGB-D histograms were trained corresponding to the probability distributions p_{obj} and p_{bg}, respectively. Furthermore, the surface area, 3D centroid and 3D covariance matrix are extracted from the projection of the ellipse into 3D space, with the information of the depth image that are used as moment constraints during segmentation.

5.4 Experimental Results

This section presents an evaluation of the approach for RGB-D image segmentation with moment constraints. The goal of the experiments was to verify that (1) segmentation on RGB-D data is more reliable than segmentation of color or depth images alone, and that (2) object tracking with 3D moment constraints is more robust than 2D moment constraints.

All images and videos shown in this chapter were captured using the Microsoft Kinect sensor. Run-times on a GPU implementation are less than 1 second per image, making the method useful for interactive applications.

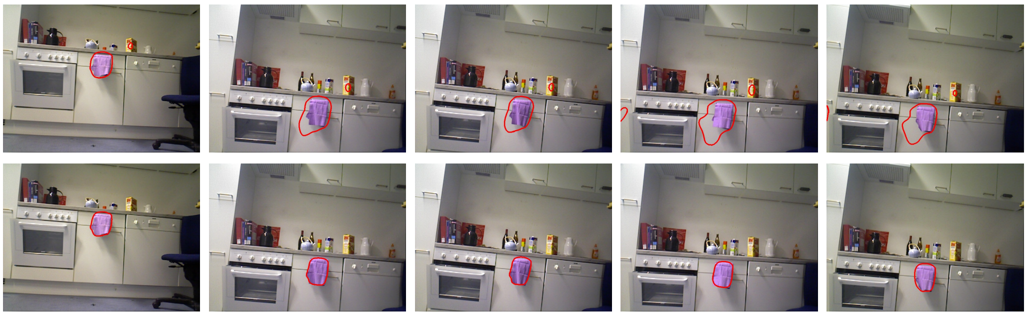


Figure 5.2: Comparison of tracking an object with and without area constraint of a scene captured from a flying quadrocopter. **First row:** Color-only tracking. **Second row:** RGB-D tracking: The surface area is constrained on the absolute dimension via additional information from the depth images.

5.4.1 Tracking in RGB-D with Moment Constraints

Fig. 5.1 shows results on moment-consistent tracking in 2D and 3D with large camera motion. In the top row we see the results for color-only tracking: The area constraint is imposed on the projected shape of the object. The method cannot cope with increasing and decreasing appearance in the image domain, although the absolute size of the object stays the same. The bottom row shows RGB-D tracking: The area constraint is imposed on the absolute dimension via additional information from the depth images. The method enables area-consistent tracking with arbitrary camera motion. The figure shows images of a plant in an office scene with a hand-held Kinect sensor from different view points. Of course, the basic properties of the 3D shape – and thus the surface area and covariance structure – of the selected object remains the same during the sequence. However, the projection of the object's shape in 2D changes its size due to object and/or camera motion. As a result, simple 2D moment tracking fails, as it tries to keep the area in image space constant. In contrast, 3D moment constraints are scale-aware and are thus more robust against camera and/or object motion. From these examples, we conclude that in the case of arbitrary camera motion 3D moment constraints are better suited for object tracking than 2D moment constraints.

The image sequence in Fig. 5.2 was captured by a flying quadrocopter with a Kinect camera mounted on top of it. The towel's shape and color distribution vary over time due to camera motion and wind caused by the quadrocopter's rotors. The figure shows that color-only segmentation (first row) is not sufficient to track the object, whereas additional information from the depth images allow 3D moment constraints to track the exact surface area

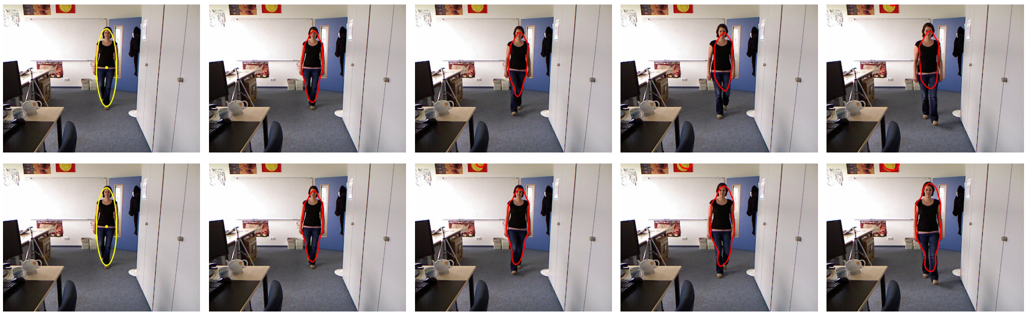


Figure 5.3: Comparison of tracking a person moving towards the camera with moment constraints in 2D and in 3D. The surface area is increased in the 2D image plane while the absolute area in 3D stays constant. **First row:** RGB tracking: Moment constraints are imposed on the projected shape in the 2D image plane. **Second row:** RGB-D tracking: The surface area, centroid and covariance are constrained on the absolute dimension via additional information from the depth images and can thus be imposed scale-aware.

(second row).

The image sequence in Fig. 5.3 shows a scene with a moving person and a fixed Kinect camera. Constraints on the first three moment constraints, i.e. the area, the centroid and the covariance are compared in 2D and in 3D. The figure shows that constraining the moments in the image plane only is not sufficient since the area the person occupies in the image changes due to the camera projection. Additional information from the depth map allows to impose the moment constraints in 3D and hence to track the absolute surface area (second row).

5.4.2 Segmentation with Color, Depth, and RGB-D

The segmentation method was tested with moment constraints in several scenes to demonstrate that RGB-D segmentation can outperform segmentation based on color or depth alone. To demonstrate this, different objects in the color, depth, and the (combined) RGB-D image were segmented.

The first example is shown in Fig. 5.4 where individual persons were to be segmented from the crowd. The figure shows that neither color nor depth information are sufficient to uniquely separate a single person in the image, see Fig. 5.4 (b+c). In more detail, the person in the first row is hard to segment in the color image because of the blue jeans in front of the blue door. The person in the second row wears a black shirt and is partially occluded by the wardrobe, and the person in the third row overlaps with the person in the background, having similar histograms which makes the segmentation

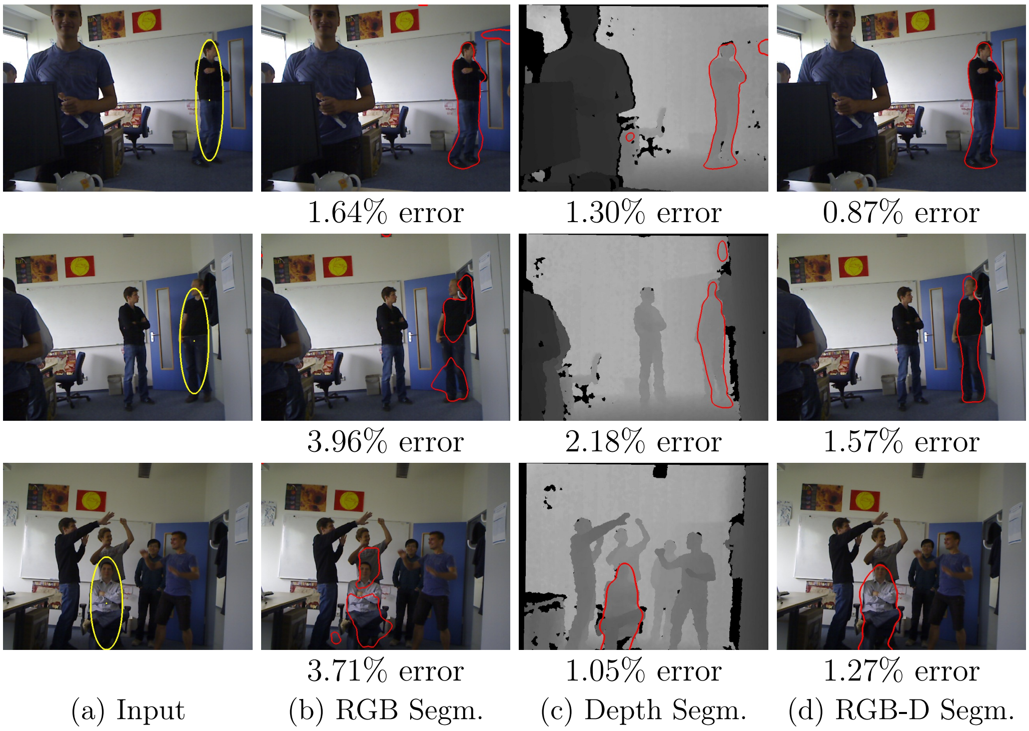
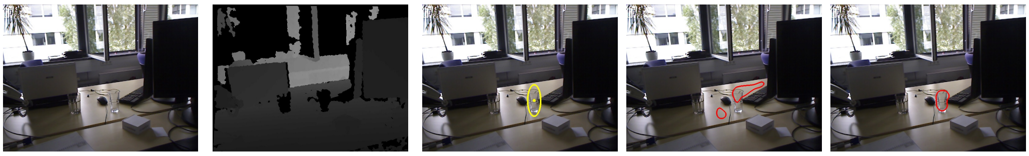


Figure 5.4: Segmentation of images with ambiguous color and depth information. Moment constraint parameters are derived from user input (a). Purely color (b) and depth (c) images alone do not provide enough information to uniquely segment one person. The combination (d) allows for segmentation of one single person in all three examples. Segmentation errors can be reduced by combining depth and color information.

task hard. Depth segmentation alone has shortcomings in other regions of the image. There are often pixels in an image with similar depth values as the foreground object – with the exception of the person sitting on the chair, where no other pixels had the same depth values. In the first two rows of Fig. 5.4, the segmentation problems are resolved when RGB and depth information is jointly considered. To conclude, all persons could be separated well in the RGB-D case.

Another interesting example is depicted in Fig. 5.5 which shows that even the absence of information in the depth image can be exploited to successfully segment an image. Here, a water glass is located on a table. In the color image, the glass is difficult to see because of its transparency. Moreover, the depth of the glass pixels cannot be estimated due to the material's reflective property. By considering both the color and the depth image, the glass is well separable. Note that the glass is not captured correctly by the depth



(a) Input Color Image (b) Input Depth Image (c) User Input (d) Color-only Segmentation (e) RGB-D Segmentation

Figure 5.5: Segmentation of reflective material with moment constraints. (a+b) Input image and (c) user input. (d) When only the color image is considered, the glass is indistinguishable from the background due to its transparency. (e) When the depth image is taken into account, the glass becomes separable.

sensor due to its reflectance. The segmentation becomes feasible due to the missing information in that area.

5.4.3 Quantitative Analysis

The quantitative analysis of the presented method shows measurements of the amount of pixels that differ from a manually segmented ground truth for segmentation with and without constraints, as well as segmentations using color, depth, and their combination. Segmentation errors were computed for the images in Fig. 5.4.

Table 5.1 shows average segmentation errors compared to the ground truths. The table also shows a comparison for segmentations without moment constraints, where segmentations were computed using only the color

		Average Segmentation Error
Without Constraints:	Color only	29.25%
	Depth only	16.99%
	RGB-D	17.93%
With Constraints:	Color only	3.10%
	Depth only	1.51%
	RGB-D	1.24%

Table 5.1: Average segmentation errors with and without moment constraints, compared to ground truth. The combination of color and depth leads to better results, even more improvement is achieved by additionally constraining the moments of the segmentation.

information of the histograms inside and outside the ellipse drawn by the user. The table clearly shows that the amount of misclassified pixels can be reduced by combining depth and color information for segmentation with moment constraints. Interestingly, segmentation with depth only yields significantly better results than color only.

5.5 Conclusion

In this chapter a convex framework for interactive RGB-D image segmentation and tracking was presented. Building on state-of-the-art approaches for color segmentation, it was shown that depth information can be integrated in the data terms for image segmentation so as to favor segmentations of coherent depth. In particular, objects of similar color but different depth can be discriminated. Moreover, it was shown that the availability of depth allows to impose constraints on the *absolute* shape rather than the *projected* shape. One can impose moment constraints in 3D space – thereby exploiting the fact that the 3D motion of a tracked object is constrained over time. The results demonstrate that combining color and depth can drastically enhance the possibilities of variational segmentation methods. In particular, it allows to generalize respective constraints from the image plane to the physical 3D space. Experiments show that with a minimal amount of user input fast interactive segmentations of good quality in a variety of challenging real-world scenarios can be obtained.

Chapter 6

Stereo Reconstruction for Phenotyping of Grapevines

This chapter describes how convex relaxation methods can be employed to compute globally optimal disparity maps. It will present a generalization of the convex formulation based on functional lifting presented in [120] with regularization based on image edge information. Image edges are an indicator for object boundaries, which is of special importance for objects with fine scaled structures as often occur in plant geometry. The chapter presents a novel method for plant phenotyping based on a convex formulation of anisotropic stereo reconstruction. The method was applied to the computation of fruit-to-leaf ratios and monitoring of grapevine growth. The growth analysis allows to identify phenotypic characteristics of a novel breeding line compared to traditional cultivars. The chapter presents a robust method for depth estimation from images that are captured directly in the field from a moving platform in a vineyard.

6.1 Introduction

Stereo reconstruction, i.e. the inference of 3D geometry from two or more 2D images, has been intensively studied in the field of computer vision. For stereo reconstruction two or more images are used that depict the same object. Respective methods can be divided in two main categories: Sparse reconstruction, as used in [69], aim at estimating single 3D points, while dense reconstruction aims at computing a surface. Sparse reconstruction methods include methods that aim to find distinctive points [99] that are easy to find in both images [132]. This leads to sparse 3D information, and especially little information in homogeneous image regions.

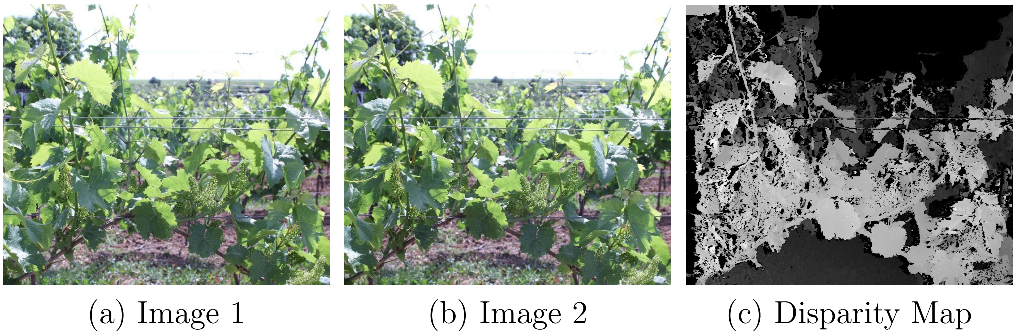

(a) Image 1 (b) Image 2 (c) Disparity Map

Figure 6.1: Disparity map reconstruction of a vineyard scene. In the disparity map the foreground plant is clearly distinguished from the background. This task would be impossible if only the color information of one image alone is used, even for a human.

In the context of field phenotyping, dense 3D surfaces are essential for a reliable reconstruction of the foreground plant. The background can contain the field as well as other plants that are farther away and have similar color distributions. Fig. 6.1 shows an example of a vineyard scene where it is hard to distinguish the foreground plant from the background using only the color information of one image. Even for the human eye a reliable segmentation of the plant is not trivial.

A representation of dense 3D information is a *depth map*, where each pixel of a reference image is assigned a distance to the depicted object point in 3D. A widely used approach for computing dense depth maps is the semi-global matching method [71]. This chapter however will focus on global optimization methods, i.e. methods that compute optimal solutions of a given energy model. Implementations include discrete optimization [72] and continuous methods [120]. Applications of stereo methods include the reconstruction of aerial images [88], driver assistance systems [121], and plant phenotyping [19, 69].

The outline of this chapter is as follows: Sec. 6.2 will describe related work on convex optimization methods for stereo reconstruction. Sec. 6.3 presents a method to integrate image edge information into the reconstruction process. In Sec. 6.4 an application of the edge-based stereo estimation method is presented for phenotyping of grapevine growth. Sec. 6.5 will give a summary of the main results.

6.2 Related Work

This section presents related work on reconstruction of depth maps. A special focus will be on the convex formulation presented in [120] which will be generalized in the subsequent section, and on different methods to compute point correspondences.

6.2.1 Convex Relaxation for Stereo Reconstruction

Dense disparity maps $v : \Omega \to \mathbb{R}$ can be computed from a pair of rectified images $I_1, I_2 : \Omega \to \mathbb{R}$. The disparity is the displacement of image locations for an object point visible in both images. Both the disparity map and the input images are defined on the same domain $\Omega \subseteq \mathbb{R}^2$. It is known from projective geometry that for a given point in one image the corresponding point in the second image lies along a line, the so called epipolar line. Hence the search space reduces to one dimension. For rectified images, the epipolar lines are parallel and axis-aligned.

A variational model for estimating a dense disparity map v is given by the functional

$$E(v) = \int_\Omega |\nabla v| \,\mathrm{d}x + \int_\Omega \rho(v(x), x) \,\mathrm{d}x, \tag{6.1}$$

The second term in (6.1), which is based on the data matching cost $\rho(v(x), x)$, is usually not convex. A globally optimal formulation for arbitrary choices of ρ, based on functional lifting, was introduced in [72] in a discrete framework and its continuous formulation in [120]. The convex method presented in [120] will be described in the following.

A Convex Formulation with Functional Lifting

Given two images $I_1, I_2 : \Omega \to \mathbb{R}$ the disparity map $v : \Omega \to \Gamma := [0, \gamma_{\max}]$ is computed in [120] by minimizing a higher dimensional variable $\phi : \Omega \times \Gamma \to [0, 1]$. A thresholded version of ϕ can be considered as an indicator function of the region enclosed by the surface to reconstruct. A detailed explanation of this concept, also called *functional lifting*, is given in [72]. The resulting disparity map v is then computed by integration over ϕ:

$$v(x) = \int_\Gamma \phi \,\mathrm{d}\gamma, \tag{6.2}$$

and ϕ is a minimizer of

$$\min_{\phi \in C} \left\{ \int_{\Omega\times\Gamma} \rho(x, \gamma) |\partial_\gamma \phi| \,\mathrm{d}x\mathrm{d}\gamma + \lambda \int_{\Omega\times\Gamma} |\nabla \phi| \,\mathrm{d}x\mathrm{d}\gamma \right\}, \tag{6.3}$$

$$C = \{\phi : \Omega \times \Gamma \to [0,1] \, : \, \phi(x,0) = 1, \, \phi(x,\gamma_{\max}) = 0\} \,, \tag{6.4}$$

where $\rho : \Omega \times \Gamma \to \mathbb{R}$ is the data fidelity term measuring the point-wise color differences between the two images. The second term is the regularizer term, weighted by a free parameter $\lambda \in \mathbb{R}$. The total variation norm as regularizer yields piecewise smooth solutions while preserving edges. Furthermore it is convex which allows for global optimization of the functional. The constraint set C ensures that the global minimum of (6.3) is not the trivial solution. This is ensured by constraining $\phi = 0$ in the disparity layer closest to the camera optical center and $\phi = 1$ in the last layer. Optimization is carried out in the three-dimensional space $\Omega \times \Gamma$ which enables convexity also in the data term. The optimization problem (6.3) can be globally optimized using a primal-dual optimization scheme [120].

The maximum disparity $\gamma_{\max} \in \mathbb{R}$ depends on the prevailing setup of the scene, i.e. the camera parameters and the distance of the camera capturing positions to each other and to the scene. When these distances are roughly known, a reliable estimate for $\gamma_{\max}$ can be determined.

Detection of Occlusions

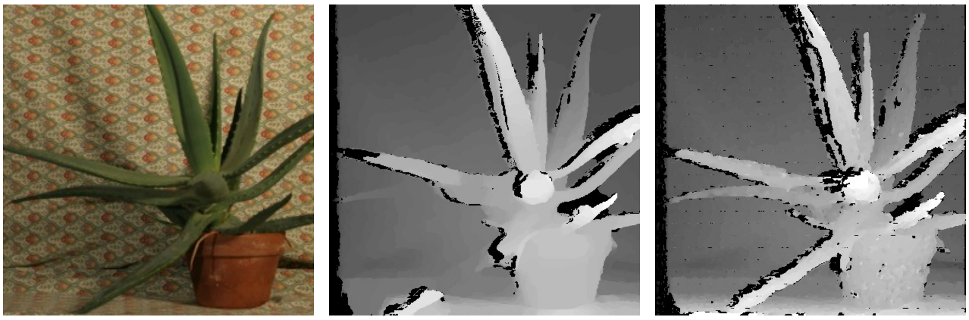
Occlusions can be detected with a *forward-backward-check* [54]. This means that disparity maps are computed in both directions, i.e. a second disparity map is computed with the left and right image I_1 and I_2 exchanged. Comparing the results provides additional information on the confidence how reliable the disparity estimate is at a given point, and allows for the detection of inconsistencies. Thresholding detects pixels where the discrepancy between the left and right disparity map is large, and assigns them the minimum disparity value $v = 0$. In the experiments shown in this chapter, the threshold was set to 20 pixels for the difference of disparities.

6.2.2 Matching Costs for Point Correspondence

In order to deduce depth information from the rectified images, pixel pairs that show the same object point have to be identified. The data term ρ in (6.3) measures color or intensity similarities between two pixels. This section will describe two of the most commonly used methods for estimating point correspondences.

Absolute Differences

The absolute difference matching cost is based on the brightness constancy assumption (BCA) which is also a basic assumption for optical flow. The



(a) Input Image (b) BCA, point-wise (c) NCC, patch-based

Figure 6.2: Disparity map reconstruction using different data terms. A patch size of 3×3 was chosen for the NCC.

BCA assumes that surfaces are *lambertian*, which implies that an object point has the same pixel color in both images. Examples for non-lambertian surfaces are shiny, transparent and reflecting surfaces.

The similarity between two intensities is measured by

$$\rho(x,\gamma) = |I_1(x) - I_2(x+\gamma)|. \tag{6.5}$$

The absolute difference is a pixel-wise matching cost that is based on the difference of color values or intensity values. If $I_1(x)$ and $I_2(x+\gamma)$ depict the same object point, the difference (6.5) should be minimal.

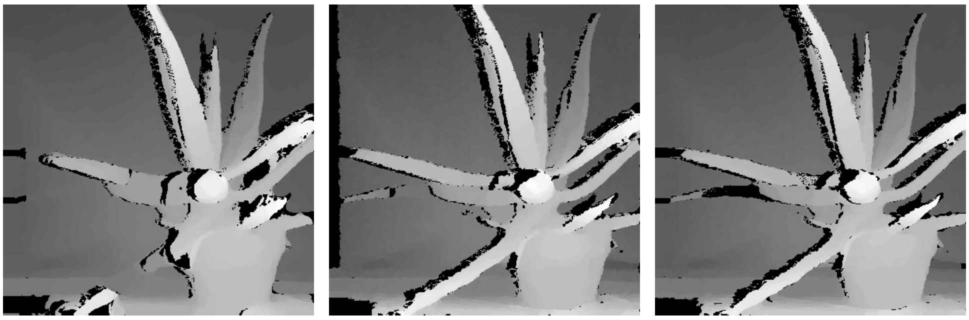
Normalized Cross Correlation

The *normalized cross correlation (NCC)* is a similarity measure for patches. It is widely used for computing matching costs in stereo reconstruction [70]. The NCC measures the similarity of the structure in a local neighborhood $\mathcal{N}$ around a point x_1 in image I_1 and a point x_2 in image I_2:

$$NCC(x_1,x_2) = \frac{\int_{\mathcal{N}}(I_1(x)-\bar{I}_1)(I_2(x)-\bar{I}_2)\,\mathrm{d}x}{\sqrt{\int_{\mathcal{N}}(I_1(x)-\bar{I}_1)^2\,\mathrm{d}x\int_{\mathcal{N}}(I_2(x)-\bar{I}_2)^2\,\mathrm{d}x}}, \tag{6.6}$$

where $\bar{I}_1$ and $\bar{I}_2$ are the mean intensities in the region $\mathcal{N}$. The NCC is invariant to additive and multiplicative intensity changes. Robustness can be increased by considering patch distortions [21] according to an estimate of the local surface geometry.

Fig. 6.2 shows a comparison of disparity maps computed with the BCA dataterm (6.5) to NCC (6.6), using an example image pair from the *middlebury* data set [126]. Other methods for computing the matching costs include the patch-based *census* and *rank* transform [147] or point-wise matching based on mutual information [71].



(a) Total variation (b) Edge weight D_1 (c) Edge weight D_2

Figure 6.3: Disparity map reconstruction using different total variation based regularization terms. Including image edges to the regularizer (b,c) yields more accurate reconstructions than non-weighted total variation (a).

6.3 Edge-Based Regularization for Stereo Reconstruction

The reconstructed disparity maps can have inaccurate shapes at object boundaries, because the region-based matching costs are defined point-wise or patch-wise. However, the edges of the input images can yield a reliable indicator for object boundaries. Incorporating image edges to disparity map estimation can be formulated using weighted and anisotropic regularizations.

Anisotropic regularization is a weighted regularization where weights are different for different directions. Applications for anisotropic regularization include data compression [127], object tracking [139] and optic flow [145].

6.3.1 Edge-Enhancing Diffusivity

The weighted total variation norm was introduced in [29] for a convex relaxation method of edge-based image segmentation. The authors suggested to weight the regularizing term with an edge detection function $g : \Omega \to \mathbb{R}$. A common choice for g in edge-based image segmentation is

$$g(x) = \exp(-\alpha |\nabla I_1(x)|) \tag{6.7}$$

with the free parameter $\alpha \in \mathbb{R}$. It yields segmentation results where the contour lines prefer the edges of the input image. In the following, a transformation of this concept to total variation based stereo reconstruction is presented.

The following variational minimization problem for stereo reconstruction weights the regularization term in x direction according to the image edges:

$$\min_{\phi\in C}\left\{\int_{\Omega\times\Gamma}\rho(x,\gamma)|\partial_\gamma\phi|\,\mathrm{d}x\mathrm{d}\gamma+\lambda\int_{\Omega\times\Gamma}\left(g(x)|\nabla_x\phi|+|\nabla_\gamma\phi|\right)\,\mathrm{d}x\mathrm{d}\gamma\right\}. \tag{6.8}$$

A similar energy model was presented in [139] for object tracking.

6.3.2 Anisotropic Diffusion Tensor

Using anisotropic regularization for functional (6.3) yields the functional

$$\min_{\phi\in C}\left\{\int_{\Omega\times\Gamma}\rho(x,\gamma)|\partial_\gamma\phi|\,\mathrm{d}x\mathrm{d}\gamma+\lambda\int_{\Omega\times\Gamma}\nabla\phi^\top D\nabla\phi\,\mathrm{d}x\mathrm{d}\gamma\right\}, \tag{6.9}$$

where the total variation norm is weighted with a diffusion tensor $D:\Omega\times\Gamma\to\mathbb{R}^{3\times3}$ [66] and the constraint set C is defined as in (6.4).

The disparity map v is defined in the image domain Ω while the diffusion tensor in (6.9) is defined on $\Omega\times\Gamma$. The following diffusion tensor corresponds to the weighted regularization in (6.8):

$$D_1=\operatorname{diag}(g(|\nabla I_1|),g(|\nabla I_1|),1). \tag{6.10}$$

A 2×2 diffusion tensor based on the image edges is given by $D=g\,nn^\top+mm^\top$, where $n=\frac{\nabla I_1}{|\nabla I_1|}$ and $m=n^\perp$ a normal vector to n. A similar diffusion tensor was presented in [145] for optical flow.

Based on these Application to stereo reconstruction yields the diffusion tensor

$$D_2=\begin{pmatrix}gn_1^2+n_2^2 & (g-1)n_1n_2 & 0\\ (g-1)n_1n_2 & gn_2^2+n_1^2 & 0\\ 0 & 0 & 1\end{pmatrix}, \tag{6.11}$$

where n_1 and n_2 are the components of n.

Fig. 6.3 shows a comparison of disparity map estimation with three different total variation based regularizations, using the same input image as in Fig. 6.2(a): a) non-weighted regularization, b) regularization weighted with diffusion tensor D_1 (6.10) and c) regularization with diffusion tensor D_2 (6.11). For the diffusivity g the term in (6.7) was used. The figure shows how the incorporation of image edge information can improve reconstruction results compared to uniformly weighted TV.

6.4 Stereo Reconstruction for Phenotyping of Grapevines

The demand for high throughput phenotyping in plant research has been increasing during the last years due to large experimental sites. Hence, time-efficient and automated processes are essential for improved field phenotyping. A major challenge for sensor based phenotyping in vineyards is the distinction between foreground (grapevine) and background (field). This section presents a method for image-based phenotyping of grapevine using RGB images captured from a moving platform in a vineyard. The method combines stereo reconstruction and image segmentation based on color and depth. The convex formulation allows for global optimization and solutions independent of initializations. The algorithms are implemented on graphics processing units (GPU) for time-efficient parallel processing.

The presented approach enables non-invasive, fast and objective estimation of plant growth. Robustness of the method is shown on images taken from a single camera directly in the field without preparing the scene. Results show efficient and robust reconstructions of surface areas. Furthermore, the images are segmented to classes corresponding to 'leaf', 'stem', 'grape' and 'background' using depth and color information. These classification facilitates determination of phenotypic indicators including the 3D leaf surface area and leaf-to-fruit ratios. The visible leaf areas of two breeding lines of grapevine were monitored during a season and used for identification of unknown growth habits and detection of differences in growth rates.

The main contribution of this section is the combination of convex relaxation methods for stereo reconstruction and image segmentation to a robust method for field phenotyping. Precise depth reconstruction and robust background subtraction enable fast image acquisition without the necessity of artificial backgrounds. Depth and color segmentations further enable objective estimation of plant growth. This advance provides a promising tool for high-throughput, fully automated image acquisition, e.g. for field robots.

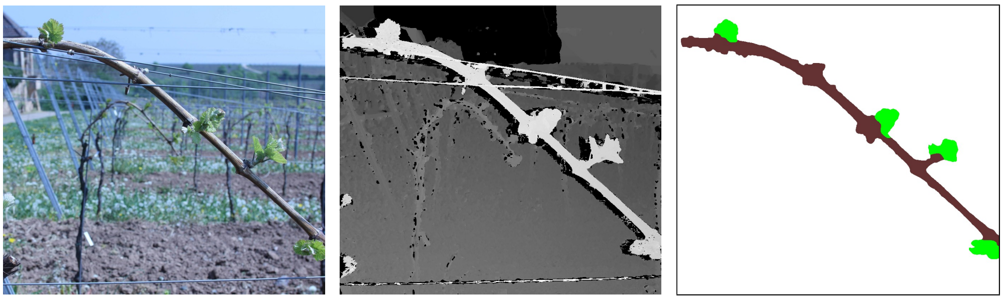
6.4.1 Non-Invasive Methods for Plant Phenotyping

Grapevines (*Vitis vinifera* L ssp. *vinifera*) are highly susceptible to several fungal diseases (e.g. powdery mildew and downy mildew) and require substantial effort in the area of plant protection. This is the major reason for extended grapevine breeding activities aiming at selection of new cultivars with both high disease resistance and high quality characteristics [138]. Due to its properties as a perennial woody crop plant, analysis of growth habits and yield traits of grapevine can only be evaluated in the field. The aim of analyzing growth habits of grapevine is to improve grape yield and wine quality [131]. Important indicators for grape quality include the geometric dimensions of canopy [101], and the ratio between vegetative (shoots and leaves) and fruit growth [131].

For the analysis of grapevine foliage directly in vineyards indirect and non-invasive methods are essential. Related methods have been presented using 3D scanners [101], ultrasonic sensors [102], LiDAR scanners [15], Greenseeker [16], plant canopy analyzer LAI-2000 [16, 40, 75] or model based strategies [98]. Other methods require destructive sampling from direct measurements with a leaf area meter [17, 76]. Active optical sensors like laser scanners directly obtain 3D point clouds of a scene, however provide no volumetric information. High-precision laser scanners as have been used in [116] are one of the most important sensors to obtain precise 3D point clouds under controlled laboratory conditions. Laser scanners for field applications are less often used because most are expensive, bulky and the data acquisition process is slow. New generation laser scanners, e.g. the Leica P20 enable a much faster acquisition of high-resolution 3D point clouds with one million points per second and a precision of up to 3 mm. However, the opportunity of scanning from a movable platform in the field is difficult due to the susceptibility against concussions. RGB-D or structured light sensors like *Kinect* are fast data capturing methods for both color and depth images. They are specialized for indoor environments, and can fail in scenes with bright illuminations or large distances [12], making them unsuitable for a usage in the field. Time-of-Flight (ToF) cameras usually have similar difficulties with sunlight [78]. For applications under controlled laboratory or greenhouse conditions these sensors are a practical tool for accurate phenotyping experiments.

Standard RGB cameras for automated analysis of grapevine growth are less commonly used. Respective methods are based on single images, which requires the application of artificial backgrounds [45], or reconstruction of sparse 3D information [122, 69].

The aim of this section is to develop a method that uses inexpensive consumer cameras capturing RGB images for reconstruction of dense depth maps. Dense 3D information can help to reduce size distortions in images when parts of the plant are closer to the camera than others. The use of a standard consumer camera enables fast data acquisition from large amounts of grapevine, and yields high-resolution color images. Being a passive optical sensor, bright sunlight does not interfere with the data capturing process, providing a practical candidate for a robust, fast and portable sensor employed directly in the field. In the following, a novel approach for non-invasive, fast and objective field phenotyping of grapevine canopy di-



(a) Input image (b) Disparity map (c) Segmentation

Figure 6.4: Image segmentation to 'leaf', 'stem' and 'background'. Depth maps are computed from pairs of RGB images, and are used in combination with color for a segmentation of the image domain. The segmentations enable objective analysis of phenotypic indicators like the visible leaf area which can be applied to growth analysis or computation of fruit-to-leaf ratios.

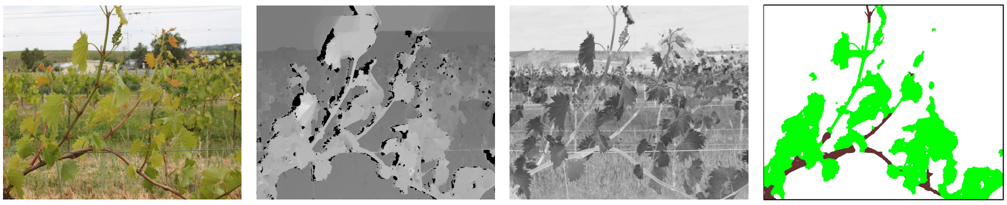
mensions is introduced. A simple setup for stereo image acquisition using a standard consumer camera is used for stereo reconstruction. The reconstruction of dense depth maps enables automated detection of grapevine in the foreground. Furthermore, image segmentation using depth and color information allows for objective quantification of visible canopy dimensions. This includes quantification of the visible leaf area, monitoring of grapevine growth and estimation of fruit-to-leaf ratios.

6.4.2 Segmentation of Grapevine using Color and Depth

90 images of grapevine plants were captured using a standard RGB camera from a moving platform at five different dates during a season. Two images of each plant were captured per date for the stereo reconstruction. The choice of 'Riesling', 'Villard Blanc' and two breeding lines was based on the similar phenology of genotypes, which implies similar time of bud burst and flowering.

Depth maps for each image pair were computed using the method described in Sec. 6.3.2. Optimization in the 3D space $\Omega \times \Gamma$ requires the respective amount of memory and run-time. The shown examples were computed on an Nvidia GeForce GTX Titan GPU using the *Cuda* programming language.

To compute the image segmentation, each pixel in the image domain Ω is assigned a label $l \in L = \{1, \ldots, n\}$. The segmentation is computed using information from both the color images and corresponding depth maps. The



(a) One of two input images (b) Reconstructed depth map (c) Green-blue color channel (d) RGB-D segmentation

Figure 6.5: (a): Input images are taken from a moving platform in the field. (b): Disparity maps are computed from the rectified input images. (c): Subtracting the blue from the green color channel yields a robust classifier for vegetation. (d): Segmentation based on color and depth.

following minimization problem computes a segmentation $u : \Omega \times L \to \{0, 1\}$ of the RGB and depth image to n regions:

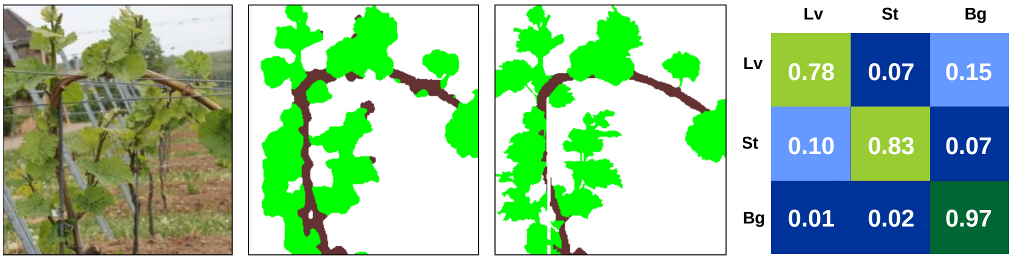
$$\min_{u \in \mathcal{B}} \left\{ \sum_{i=1}^{n} \left(\int_\Omega f_i(I_1^{\text{green}} - I_1^{\text{blue}}, d) u_i \, \mathrm{d}x + \nu \int_\Omega |\nabla u_i| \, \mathrm{d}x \right) \right\}, \text{s.t.} \sum_{i=1}^{n} u_i = 1. \tag{6.12}$$

A convex minimization problem is obtained when u is relaxed to continuous functions and can then be solved globally optimal [119]. The data terms $f_i : \Omega \times L \to \mathbb{R}$ are computed using the depth map $d(x)$ and color image $I_1(x)$ (the reference image of the stereo pair). It is based on the assumption that the background is farther away from the camera capturing position than the foreground plant. The foreground is further divided to 'leaf' and 'stem'. Subtracting the blue from the green color channel has been shown to be a robust classifier for vegetation, since it enhances green regions [105]. For an example of a green-blue color channel see Fig. 6.5 (c).

The experiments showed that a two step approach yields more accurate results than combining the background subtraction and segmentation of plant components. The data terms are based on two parameters $c_{\text{depth}} \in \mathbb{R}$ and $c_{\text{color}} \in \mathbb{R}$ that are related to the maximal distance of the plant to the camera and the saturation of the green of the leaves, respectively. First, the image domain is segmented using the reconstructed depth map d. The function

$$f_1(x) = d(x) - c_{\text{depth}} \tag{6.13}$$

implements the assumption that the background is farther away from the camera capturing position than the foreground plant. The parameter c_{depth} should be set to the maximum depth that the foreground plant can obtain.



	Lv	St	Bg
Lv	0.78	0.07	0.15
St	0.10	0.83	0.07
Bg	0.01	0.02	0.97

(a) Input image (b) Segmentation (c) Ground truth (d) Confusion matrix

Figure 6.6: Evaluation of classification results. (a): Input image (b): Computed segmentation (c): Reference segmentation from a manually labeled image (d): Confusion matrix for comparing both types of classification results. Computed segmentations are shown in rows and reference segmentations in columns. Lv - ’Leaf’; St - ’Stem’; Bg - ’Background’.

It can be assumed constant for standardized image capturing processes, or if distances of the camera capturing positions and plants vary only in a specified range.

Second, the foreground is segmented to ’leaf’ and ’stem’ based on the difference of the green and blue color channel:

$$f_2(x) = I_1^{\text{green}}(x) - I_1^{\text{blue}}(x) - c_{\text{color}}. \tag{6.14}$$

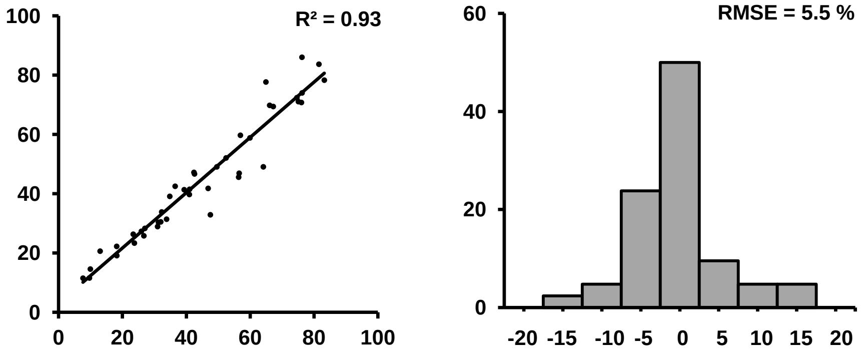
The parameter c_{color} is mainly dependent on illumination and weather conditions of the scene. In the experiments shown in this chapter, $c_{\text{color}} = 0.08$ was determined as a suitable value, when RGB values range from 0 to 1.

6.4.3 Evaluation and Error Analysis

The major aims of this study were 1) a background subtraction from field images using the reconstructed disparity maps, 2) a segmentation of the visible leaf area, and 3) a quantification of leaf areas to enable objective phenotyping of grapevine growth.

Ten images and 32 randomly selected image sections were analyzed to compare the segmentation results to manually segmented ground truths. Fig. 6.6 shows an example for an input image (a), the computed segmentation (b) and a manually segmented ground truth (c). The confusion matrix (d) shows that the major percentage of the three regions was correctly classified, while best results were obtained for the background subtraction with 97 % of correct classifications.

For the monitoring experiment, standard deviations of leaf areas of the reference cultivars ’Riesling’ and ’Villard Blanc’ were computed. From the



(a) Linear regression analysis (b) RMSE

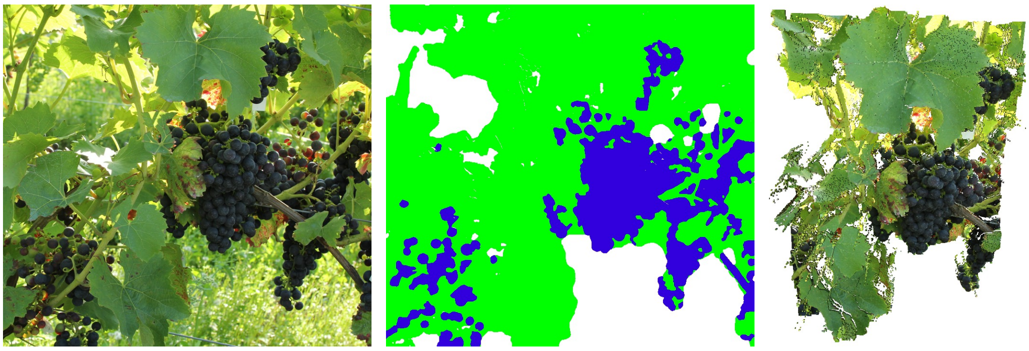
Figure 6.7: Error analysis of the reconstructions. (a): Linear regression analysis (segmentation results versus ground truth). (b): Frequency distribution of the observed residue and Root-Mean-Squared-Error (RMSE) (residuum versus frequency in %).

breeding lines only one plant per genotype was available and thus, no genotype specific variations are examinable at a time point. The regression analysis showed an R^2 coefficient of determination R^2=0.93 (Fig. 6.7 (a)). This implies that the regression line approximately represents the reference data. As shown in Fig. 6.7 (b), the residue has a normal frequency distribution and a Root-Mean-Squared-Error of RMSE = 5.5 %. The figure shows that 50 % of the predicted data shows a residuum between -2.5 and 2.5 %. 17 % of the data points shows a residuum greater than -/+7.5 %.

6.4.4 Reconstruction of the Visible Leaf Surface Area

Visible leaf surface areas are used to objectively evaluate yield efficiency. The segmented images can be used to compute surface areas of the respective regions. With additional 3D information, absolute surface areas can be computed. This allows for a scaling of pixel sizes according to their depths. This can balance out the fact that some parts of the scene are closer to the camera and thus receive a disproportionally larger area in the projected 2D plane than the grapes that are farther away.

The 3D visible leaf surface area is estimated from the classified images and the dense depth maps. It is computed by weighting the pixel sizes according to their depth. Using the depth maps, the projection effect can be compensated by computing the relative sizes of the depicted object parts to each other. The 3D surface area of region Ω_i is computed from the segmentation



(a) Input Image (b) Segmentation (c) 3D Surface

Figure 6.8: Segmentation of grapes for estimation of fruit-to-leaf ratios. The input image (a) is segmented to regions corresponding to 'grape', 'leaf' and 'background' (b). The actual ratio of grapes per leaf can be determined using either the 2D image domain or the computed 3D surface (c).

u and depth map d by integrating the point-wise areas in Ω_i:

$$\mathrm{Area}(\Omega_i) = \int_\Omega d(x)^2 u_i(x)\,\mathrm{d}x, \tag{6.15}$$

where the size of a pixel is computed as $d(x)^2$, normed by the focal length f of the camera, as shown in Sec. 5.3.2. If a reference measurement like the baseline is given, the absolute area can be computed, otherwise it can be computed up to a constant factor. For applications where relations between areas are computed like fruit-to-leaf-ratios, the absolute scale has no effect on the result because it gets factored out.

6.4.5 Estimation of Fruit-to-Leaf Ratios

The phenotypic class 'grape' was implemented to estimate fruit areas. The area of the 'grape' and the 'leaf' regions were computed in two different domains: 1) in the two-dimensional image plane (Fig. 6.8 (b)), and 2) in the three-dimensional space, using 3D information from the computed depth maps (Fig. 6.8 (c)). For 3D estimation, a baseline of 1 m was assumed. For computations in 3D, actual sizes were estimated for each class in cm^2. Both 2D and 3D data were used for the computation of fruit-to-leaf ratios. Tab. 6.1 shows a comparison of the grapes-to-leaf ratio in 2D (%) and 3D (cm^2). In comparison to the results computed in 2D the leaf area is increased by 10 % when additional 3D information is considered. Accordingly, this results in a 10 % decreased fruit-to-leaf ratio.

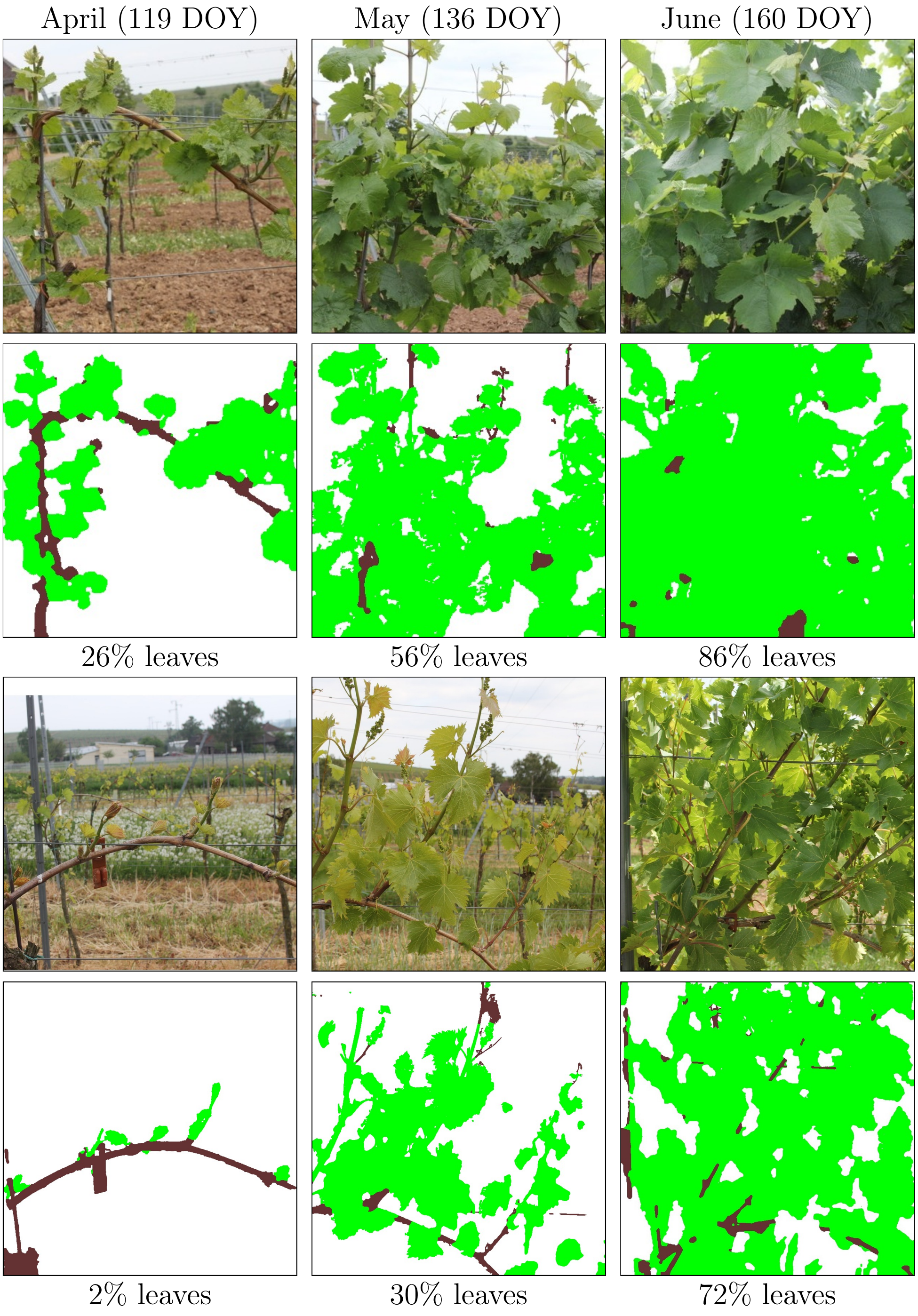
Area in 2D:	42.61 % leaves 26.07 % grapes	Grape-to-leaf ratio in 2D: 0.61
Area in 3D:	1725.45 cm^2 leaves 878.02 cm^2 grapes	Grape-to-leaf ratio in 3D: 0.51

Table 6.1: Computation of fruit-to-leaf ratios in 2D and 3D. The depth weighted 3D space allows for more accurate results than the 2D image plane.

6.4.6 Monitoring of Grapevine Growth for Improved Field Phenotyping

An important application of the presented method is the monitoring of growth habits of breeding material with unknown properties in comparison to traditional cultivars used as a reference. Fig. 6.9 shows some of the images that were used for monitoring. They show two cultivars at different time points during a season. Fig. 6.10 shows the progression of leaf area per breeding line and average leaf areas of the traditional cultivars. For all genotypes, an increasing leaf area was measured between the 90th day and the end of the experiment on day 160. These differences in leaf areas were used for objectively scoring the plant growth. As also shown in Fig. 6.10, the genotypes offer the major differences in plant growth at day 120, although 'Villard Blanc' showed only a minor increase in leaf area. Another two weeks later two groups were observed: group 1 ('Riesling' and Breeding line 1) and group 2 ('Villard Blanc' and Breeding Line 2). At day 160, breeding line 1 almost displayed the maximum feasible leaf area of 100 %. This genotype also exhibited the fastest growth during the entire experiment. The second breeding line grew at a slower rate and had a smaller leaf area at day 160 and thus seems to be more related to 'Villard Blanc'. Furthermore, the data was used for a comparison of growth rates. The figure shows that the average leaf area of 'Riesling' rises rapidly from 0 % (104th day) to 26 % (119th day). In contrast, 'Villard Blanc' shows a growth rate of only 2 % in average at the same time. Approximately 40 days later, average leaf areas of 89 % for 'Riesling' and 67 % for 'Villard Blanc' were determined. Both cultivars showed an increase of the average leaf area of approximately 65 %.

'Riesling' shows a ten times faster growth compared to the cultivar 'Villard Blanc'. In addition, two groups of growth habits can be observed, which enable an objective evaluation of the investigated breeding lines. Thus, the presented method provides a promising tool for the identification of e.g. genotype specific differences in growth rates or for the investigation of the efficiency of plant protection efforts. This kind of fast, objective and compar-



26% leaves 56% leaves 86% leaves

2% leaves 30% leaves 72% leaves

Figure 6.9: Image-based monitoring of grapevine growth. **First and second row**: Increasing leaf area of a 'Riesling' (medium shoot growth) from April (26 %) to June (86 %). **Third and fourth row**: Increasing leaf area of a 'Villard Blanc' (weak shoot growth) from April (2 %) to June (72 %).

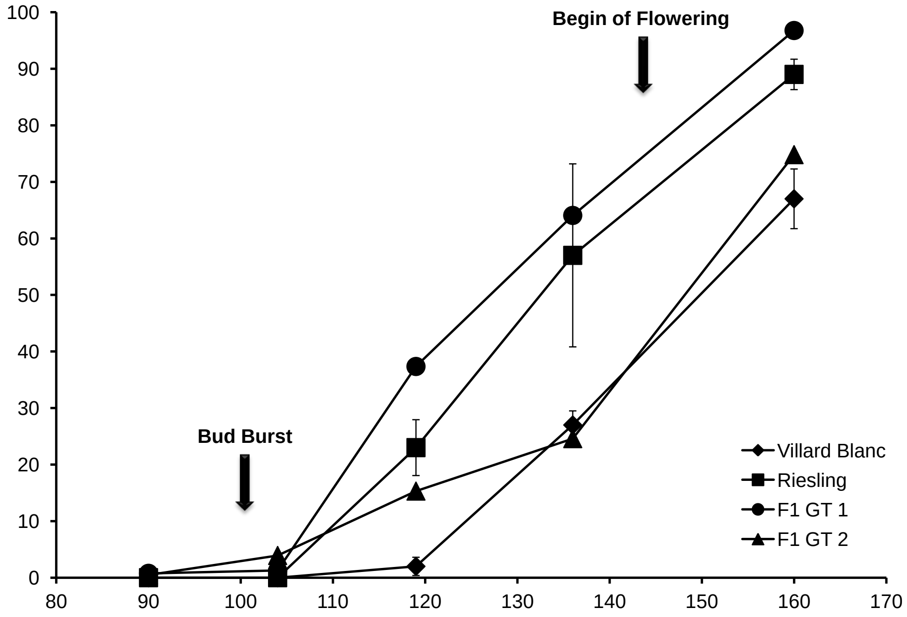


Figure 6.10: Visible leaf areas for different grapevine cultivars (day of the year versus computed leaf area in %). Differences in leaf area allow for monitoring of vegetative growth. Arrows denote the day of the year when bud burst and begin of flowering was scored (reference evaluations). Mean values and standard deviations are shown for 'Riesling' and 'Villard Blanc'. Two breeding lines with unknown growth characteristics were compared to traditional cultivars.

ative monitoring of plant development further enables the study of growing dynamics with regard to climatic influences or soil properties.

6.5 Conclusion

This chapter has described how convex relaxation techniques can be used for stereo reconstruction using functional lifting. An extension of the method to anisotropic regularization was presented, which results in depth reconstructions whose discontinuities are better aligned with the image edges compared to total variation regularization. It was shown that the consideration of image edges indicated by the gradient norm can significantly improve reconstruction results. Anisotropic weighting in image direction by an edge detector was shown as an efficient way to obtain more accurate disparity maps.

Furthermore an image-based approach for field phenotyping of grapevine growth was presented. The method uses RGB image pairs captured in unprepared outdoor environments using a standard consumer camera. Images of the same plants were captured at different dates during a season and were processed for growth analysis. The ability to accurately and quickly monitor phenotypic plant growth, particularly after bud burst, facilitates an improvement to vineyard management, and the early detection of growth defects. The method provides a notable advance and a promising tool for high-throughput, fully automated image acquisition, e.g. by using field robots.

The next chapter describes a method for multi-view stereo reconstruction which also can be applied to plant phenotyping. Whereas in this chapter a high-throughput method for reconstruction of depth maps in difficult outdoor situations was presented, the next chapter focuses on highly detailed, full 3D models.

Chapter 7

Multi-View Stereo Reconstruction with Convex Relaxation

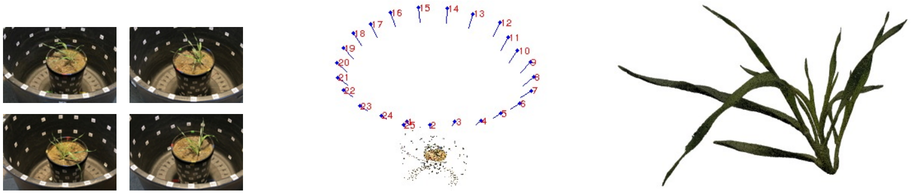
This chapter presents a globally optimal 3D geometry reconstruction method based on the convex relaxation method for image segmentation described in Chapter 3. Furthermore this chapter presents a run-time and memory-efficient implementation with octrees that is specialized to high-resolutions and is thus suitable to reconstruct objects with thin structures. Volumetric 3D models are computed in a convex optimization framework from a set of RGB input images depicting the object to reconstruct from different view points. Results show accurate 3D models, while an increase in resolution of a factor of up to 2000 is achieved in comparison to the use of a uniform voxel based data structure, making the choice of data structure crucial for feasible resolutions. Since thin structures typically occur in plant geometry, an application of the presented method to plant shape reconstruction is presented.

Parts of this chapter have been published in [4, 8, 9, 10].

7.1 Introduction

Reconstructing the 3D geometry of a scene from a set of images is one of the most studied problems in computer vision. The general formulation has attracted a variety of researchers to tackle the problem. The problem of estimating the 3D structure of a scene from a collection of 2D projections in images is a so-called *ill-posed* problem. *Ill-posed* in this context means that the solution is not unique. This is due to the fact that the projection process during the image capturing reduces the dimension by 1.

Variational methods have been established as the preferred method be-



(a) 4 of 25 Input Images

(b) Camera Positions

(c) Reconstructed 3D Model

Figure 7.1: Volumetric 3D reconstruction using an octree data structure. The reconstructed 3D model was computed from 25 input images and consists of ∼12 million octree nodes.

cause they provide a well defined mathematical concept with provable optimization and convergence guarantees. Efforts have been made using level sets [49, 57], triangle meshes [42], graph cuts [90] and convex formulations [9, 64]. Some of these concepts will be discussed in Sec. 7.2.

This chapter presents a convex optimization method for volumetric 3D reconstruction from a set of RGB images, as well as an extension specialized on accuracy and high-resolution. The method is implemented in a convex framework allowing for global optimization of the chosen model which makes it independent of initializations. The underlying data structure is based on octrees, which enable a fast and memory-efficient implementation, making high resolutions possible. The experiments show that the choice of data structure is not only beneficial for reducing run-time and memory requirements, but crucial to make high-resolutions possible.

Fig. 7.1 shows results of the proposed method for a 3D reconstruction from 25 images. The use of octrees enables a more than 2000 times higher resolution compared to a uniformly spaced voxel grid using the same amount of memory. Especially in the case of thin structures, the data structure is critical to avoid memory limitations.

The outline of this chapter is as follows: Sec. 7.2 will give a short overview of related work in the field of multi-view stereo reconstruction from images as well as memory-efficient implementations. In Sec. 7.3 a convex formulation for multi-view stereo reconstruction is described that allows for continuous global optimization. Sec. 7.4 presents a memory-efficient implementation for volumetric reconstruction of thin structures based on the octree data structure. Sec. 7.5 will describe how the method can be useful for plant phenotyping, using the example of barley. Sec. 7.6 will give a summary of the main results.

7.2 Related Work

Convex optimization methods provide a powerful technique for inferring the 3D structure of an object from a set of images in a globally optimal way [9]. Volumetric methods as used in [9, 64] allow for reconstructions of dense surfaces, at limited resolution due to large memory requirements of the underlying data structures. Point cloud reconstructions from images as used in [55] require less memory while neglecting density. 3D reconstruction based on minimizing the reprojection error has been presented for triangle meshes in [42] and for level sets in [57]. An early work on variational methods for 3D stereo reconstruction is the *stereoscopic segmentation* presented in [146].

3D reconstruction of thin structures requires special consideration to the fine scaled features. The usual assumption that the object to reconstruct is compact does not apply. Reconstruction of thin structures based on silhouette constraints as proposed in [38] allows to preserve fine structures while the uniform voxel based data structure still limits the resolution. For thin objects volumetric approaches yield large amounts of empty space, implying the need for more efficient non-uniform data structures.

Originally introduced for computer graphics, octrees [103] provide a memory-efficient data structure for large scale 3D objects. Large-scale reconstructions for fusion of RGB-D images into a volumetric model have shown that an octree based data structure avoids memory limitations in 3D reconstructions [133].

A non-hierarchical memory-efficient approach for volumetric representations is the narrow band method. The narrow band method was introduced in 1995 for volumetric 3D reconstruction with level sets [13]. Narrow band methods allow for run time improvements by optimizing the model inside a narrow band of a current estimate of the surface, instead of the whole volume. Depending on the implementation, a memory reduction can also be achieved. Narrow bands for 3D reconstruction in graph cuts have been presented in [90].

7.3 A Convex Formulation for Multi-View Stereo Reconstruction

This section presents a volumetric method for 3D reconstruction from a set of images. Volumetric methods optimize an implicit representation of the surface to reconstruct inside a partitioning of a bounding box into a grid of voxels. A *voxel* is a small volumetric element and can be considered as the three dimensional extension of an pixel. Volumetric methods try to estimate

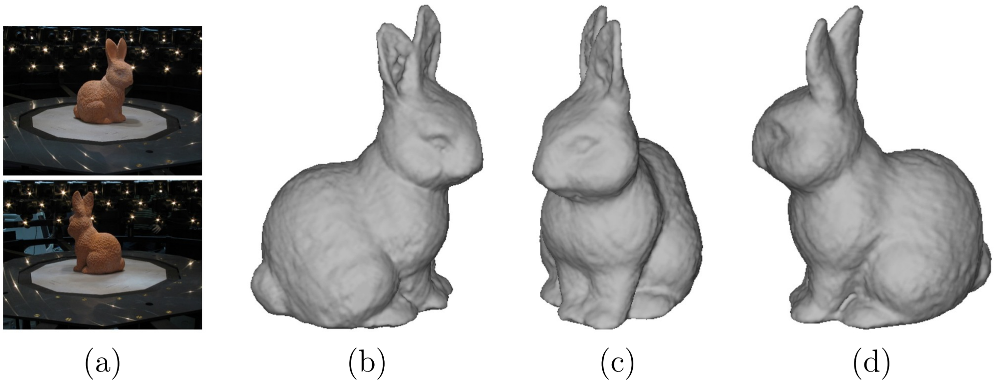


Figure 7.2: Volumetric 3D reconstruction of a rabbit figure, computed from 33 input images. The resolution of the model is $216 \times 288 \times 324$.

for each voxel if it belong to an object or the empty background space. The surface to recover is then defined as the boundary of the set of voxels belonging to an object. For an example see Fig. 7.2.

We consider a continuous image domain $\Omega \subset \mathbb{R}^2$. Given a set of m input images $I_1, \ldots, I_m : \Omega \to \mathbb{R}$ depicting the object from different view points, a surface $\Sigma \subset \mathbb{R}^3$ is computed that gives rise to the images. To reconstruct a full 3D model, each object point must be visible in at least two images. Fig. 7.1 (b) shows an example for 25 camera positions, computed with software of [99] and [132].

7.3.1 Variational Surface Reconstruction

A variational method for 3D reconstruction in level sets was presented in [49]. Minimizing the energy model

$$E(\Sigma) = \int_\Sigma g(s)\,\mathrm{d}A \tag{7.1}$$

yields a weighted minimal surface problem. Here $g : V \to \mathbb{R}^+$ measures a photoconsistency of the 3D object point. Photoconsistency measures the similiarity of the object point projected to the input images. It can be computed using one of the methods described in Sec. 6.2.2.

7.3.2 Convex Relaxation

For global optimization, functional (7.1) needs an additional regional data term, because the global minimum of (7.1) is the empty set. The following non-convex minimization problem is based on the surface Σ and regional data terms for the inside and outside regions:

$$\min_{\Sigma} \left\{ \int_{\Sigma} g(s) \, \mathrm{d}A - \nu \int_{int(\Sigma)} f_{obj}(x) \, \mathrm{d}x - \nu \int_{ext(\Sigma)} f_{bck}(x) \, \mathrm{d}x \right\} . \qquad (7.2)$$

It consists of a surface based data term g and regional data terms f_{obj} and f_{bck}. The first term integrates the surface Σ, weighted with a photoconsistency measure $g : V \to \mathbb{R}$.

A global optimization method for volumetric multi-view stereo reconstruction was presented in [9]. Convex relaxations of (7.2) are achieved by introducing an implicit representation of the surface Σ with $u : V \to [0, 1]$, defined on a volume $V \subseteq \mathbb{R}^3$. The surface Σ is represented implicitly by an indicator function $\mathbf{1}_\Sigma : V \to \{0, 1\}$ that defines a segmentation of the volume V to object, i.e. $\mathbf{1}_\Sigma(x) = 1$, and background, i.e. $\mathbf{1}_\Sigma(x) = 0$. Relaxing the range to the continuous range $[0, 1]$ allows for convex optimization of the corresponding segmentation $u : V \to [0, 1]$. The following minimization problem is a convex formulation of (7.2):

$$\min_{u \in \mathcal{B}} \left\{ \int_V g(x) \, |\nabla u| + \nu \int_V f(x) u(x) \, \mathrm{d}x \right\} , \qquad (7.3)$$

where f is the data term measures the regional terms for the foreground and background region:

$$f(x) = f_{obj}(x) - f_{bck}(x). \qquad (7.4)$$

The thresholding theorem that was shown in [34] for a convex formulation of image segmentation, applies here as well, as has been shown in [9].

7.3.3 Surface Optimization

The global minimium of (7.3) can be computed via a gradient descent or SOR scheme as shown in [9]. The Euler-Lagrange equation is a necessary condition for a minimum of (7.3) and gives rise to the gradient descent update scheme:

$$0 = \nu \operatorname{div} \left(g \frac{\nabla u}{|\nabla u|} \right) - f. \qquad (7.5)$$

Fig. 7.3 shows reconstruction results for the dino from the *middlebury* data set [126].

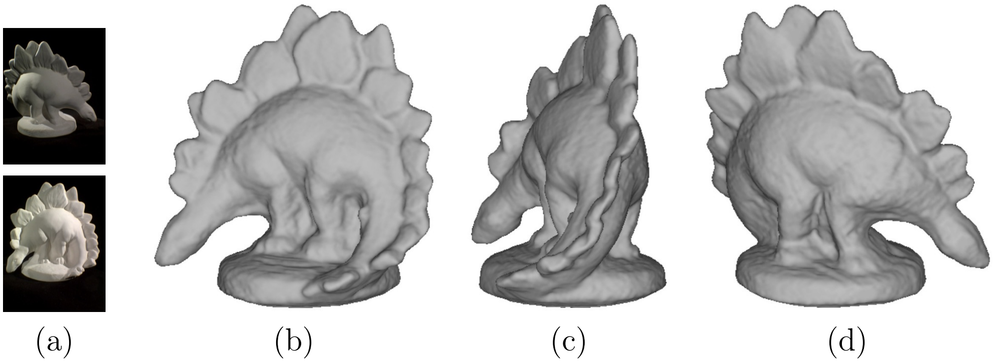


Figure 7.3: Volumetric 3D reconstruction of a dino figure, computed from 48 input images of the *middlebury* data set.

7.4 Multi-View Stereo Reconstruction with Octrees

For reconstruction of thin structures, the volumetric approach reaches its limits due to the limited resolution of the voxel grids. Increasing the resolution reaches the feasible amount of memory even for relatively low resolutions as will be shown in Sec. 7.4.3. Furthermore a large set of empty voxels is wasted in the background, that can be merged to larger blocks of voxels needing less memory. This implies the advantage of hierarchical data structures for the volumetric approach. In this section, octrees as the underlying data structure are proposed for accurate reconstructions of thin structures.

7.4.1 Surface Optimization with Volume Constraints

The surface is optimized inside the visual hull [92] $\mathcal{H} \subset \mathbb{R}^3$ which is determined by silhouette images. The silhouette images $S_i : \Omega \to \{0,1\}, i = 1, \dots m$ are defined as $S_i(p) = 1$ at points $p \in \Omega$ that depict the plant and $S_i(p) = 0$ otherwise, i.e. at points that depict background. The silhouette images can be computed using an interactive image segmentation method like the one presented in [140]. The visual hull is the smallest volume whose projections to the input images cover the silhouettes of the object.

Additional volume constraints ensure a stable substance of the reconstructed object. Volume constraints have been used for example in single view reconstruction [137] and image segmentation [3, 62].

The surface Σ is represented implicitly by an indicator function $\mathbf{1}_\Sigma : \mathcal{H} \to \{0,1\}$ that defines a segmentation of the volume enclosed by the visual hull $\mathcal{H}$. Relaxation to the continuous range $[0,1]$ allows for convex optimization of the corresponding segmentation $u : \mathcal{H} \rightarrow [0, 1]$.

We consider the following convex optimization problem

$$\min_u \left\{ \int_{\mathcal{H}} g(x)\,|\nabla u| + \nu \int_{\mathcal{H}} f(x)u(x)\,\mathrm{d}x \right\}, \quad \text{s.t.} \quad \mathcal{V}(u) \geq c, \tag{7.6}$$

where $\mathcal{V}$ refers to the volume of the object, i.e.

$$\mathcal{V}(u) = \int_{\mathcal{H}} u(x)\,\mathrm{d}x. \tag{7.7}$$

and $c \in \mathbb{R}$ is the minimum volume. In the experiments shown in this chapter, the volume constraint parameter was set to $c = 0.9 \cdot |\mathcal{H}|$ which implies that the volume of the segmented object should be at least 90% of the volume enclosed by the visual hull. The data term $f : \mathcal{H} \rightarrow \mathbb{R}$, weighted with $\nu \in \mathbb{R}$, implements the assumption that the visual hull is a rough estimator for the object and is based on the distance of a point to the border $\partial\mathcal{H}$ of the domain:

$$f(x) = 1 - \min_{\hat{x} \in \partial\mathcal{H}} \parallel x - \hat{x} \parallel . \tag{7.8}$$

For the m input images $I_1, \ldots, I_m : \Omega \rightarrow \mathbb{R}$ the photoconsistency g is computed as the intensity difference of the best matching image pair:

$$g(x) = \min_{i,j \in \{1,\ldots,m\}, i \neq j} |I_i(\Pi_i(x)) - I_j(\Pi_j(x))|, \tag{7.9}$$

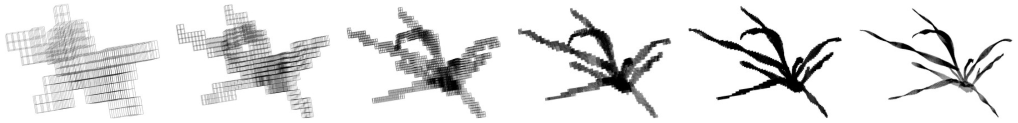
where $\Pi_i : \mathbb{R}^3 \rightarrow \Omega$ is the projection of a 3D point x to image I_i. The photoconsistency term $g(x)$ is used as a weighting function for the gradient norm $|\nabla u|$ to direct the surface through points whose projections to the images have similar intensity values. This formulation of the photoconsistency assumes that the surface to reconstruct is *Lambertian*, i.e. not reflecting or translucent.

A minimizer of (7.6) is computed using a primal-dual optimization [33] scheme with gradient descent in the primal variable u and gradient ascent in the dual variable $p : \mathcal{H} \rightarrow \mathbb{R}^3$

$$p^{t+1} = \pi_C \left(p^t + \tau_p \nabla u^t \right) \tag{7.10}$$
$$u^{t+1} = \pi_{\mathcal{V}} \left(u^t + \tau_u (\mathrm{div}(p^{t+1}) - \nu f) \right) \tag{7.11}$$

and the projections are given by

$$\pi_C(p) = \frac{p}{\max\left\{1, \frac{|p|}{g}\right\}} \tag{7.12}$$



Depth 7 Depth 8 Depth 9 Depth 10 Depth 11 Depth 13

Figure 7.4: Octree data structure for increasing resolution, i.e. depth of the tree. The figures show the bounding boxes of all leaf nodes in the deepest level of the octree. The data structure is built in a top-down method where each level is complete in itself but can be refined for higher resolution.

$$\pi_{\mathcal{V}}(u) \quad = \quad u + \max\left\{\frac{1}{|\mathcal{H}|}\left(c - \int_{\mathcal{H}} u(x)\,\mathrm{d}x\right), 0\right\}. \tag{7.13}$$

The time steps τ_p and τ_u were set to $\tau_p = \tau_u = 0.3$. The projection π_C was presented in [31]. The projection $\pi_{\mathcal{V}}$ projects the current u to the volume constraint $\mathcal{V}(u) \geq c$, and is computed analogue to the area constraint in [3]. The boundary conditions are Dirichlet conditions for the gradient, i.e. $\nabla u|_{\partial\mathcal{H}} = 0$, and Neumann conditions for the divergence, i.e. $\operatorname{div}(p)|_{\partial\mathcal{H}} = p$.

7.4.2 A Memory-Efficient Data Structure using Octrees

An octree is a tree data structure whose nodes have either eight or no sub nodes. Nodes with eight sub nodes are denoted as *inner nodes* and nodes without sub nodes as *leaf nodes*. Octrees provide a memory-efficient data structure for 3D volumes.

Building the Octree

The octree data structure is computed from the silhouette images in a top-down approach starting at a root node enclosing the whole scene depicted in the images. Subsequently, nodes are subdivided depending on the structure of the visual hull. Fig. 7.4 shows an example octree at different steps of the iteration.

Each node gets a assigned a bounding cuboid with coordinates $C := (x_{\min}, y_{\min}, z_{\min}, x_{\max}, y_{\max}, z_{\max})$ that define the volume enclosed by the node. The camera positions and viewing angles define a bounding cuboid which define the respective coordinates of the root node. The nodes are subsequently divided into eight sub-nodes of equal size if the visual hull passes the bounding cuboid of the node. The visual hull passes the cuboid if the projection of the cuboid's faces to the images contains both plant and background for at least one of the m input images. The nodes are refined until a predefined maximal

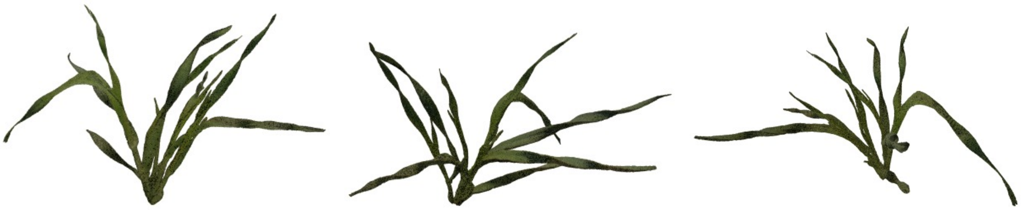


Figure 7.5: Volumetric 3D Reconstruction of a barley, computed from 25 RGB images. The dense surface is optimized in the leaf nodes of deepest level in an octree of depth 14.

depth is reached that corresponds to the desired resolution. In each iteration the octree contains the visual hull in the leaves of the deepest level. Note that it is not necessary that the bounding cuboid is as small as possible since the subsequent subdivision of the data structure will prevent the allocation of too many nodes.

Neighborhood Connectivity of Nodes

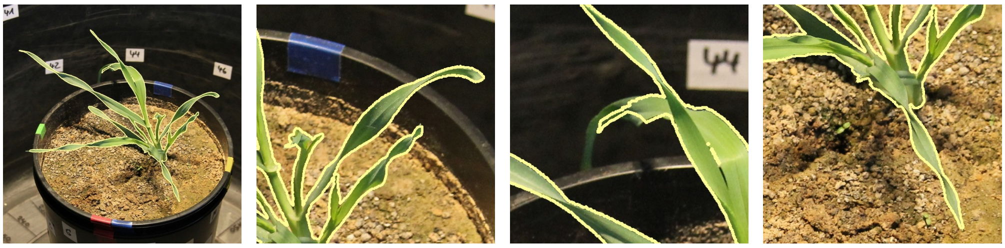
To compute the derivatives for the gradient and divergence operators in the optimization update steps (7.10) and (7.11) each leaf node in the octree requires access to the function values of its neighboring nodes. Each node stores a reference to its parent node, and the inner nodes also to the eight sub nodes. Storing additional references to the six neighboring nodes respectively saves run-time while needing more memory. The neighboring nodes are computed for each node every time when access to it is needed. We chose not to precompute them, because experiments showed that the run-time improvement is not significant. Due to the bounding cuboid each node has defined, neighboring nodes can be found by its coordinates via traversing one path of the tree from the root to the node. The respective run-time is in $O(\log(n))$, where n is the number of nodes in the octree and $\log(n)$ is the maximal depth.

7.4.3 Performance Evaluation

The method is evaluated with respect to accuracy and memory requirements for 3D reconstructions of barley.

High-Resolution Volumetric 3D Reconstruction

Fig. 7.5 shows reconstruction results for barley for the input images shown in Fig. 7.1(a). The images were captured with a standard consumer camera



(a) Silhouette projection

(b) Close-up view 1

(c) Close-up view 2

(d) Close-up view 3

Figure 7.6: The high-resolution data structure allows for accurate 3D reconstruction. (a): The silhouette of the reconstructed 3D model is projected to one of the input images. (b-d): Close-up views visualize the accuracy of the reconstruction. The similarity of the projected silhouette compared to the ground truth is 0.96.

at a resolution of 5184 × 3456 pixels. The camera capturing positions were computed using the software of [99] and [132]. The octree that was computed to reconstruct the 3D model has a depth of 14 and its computation took around 30 minutes, making the method suitable for off-line reconstructions. For a plant of 10 cm height a resolution of $1.8 \cdot 10^{-6}$ mm^3 is achieved by the use of the octree data structure.

Accuracy of the Reconstruction

The accuracy of the reconstructed 3D model is measured by projecting its silhouette to the input images and computing the difference to manually segmented ground truth images. Since an objective ground truth in 3D is not available, the projection error is measured in the image domain. Fig. 7.6 shows that the proposed 3D reconstruction with octrees enables accurate 3D reconstruction of fine-scaled structures of the plant. The figure shows a projection of the reconstructed object to one of the original images. The similarity of the projected silhouette compared to the manually segmented ground truth is 0.96. As similarity measurement the dice coefficient was used, where a value of 1 corresponds to a perfect overlap and 0 to no overlap. The close-up view in Fig. 7.6 (c) shows an example where the reconstructed model is inaccurate: the reconstruction does not contain the whole leaf in the middle of the image. In this case this is due to the fact that the leaf is not visible in some of the images and the region is hence segmented as background in 3D.

	Uniform Grid:		Octree:		
Octree Depth	Number of Voxels	Memory Requirement	Number of Nodes	Memory Requirement	Comparison Factor
7	64^3	1 MB	10^3	85 KB	12
8	128^3	8 MB	$3 \cdot 10^3$	240 KB	34
9	256^3	64 MB	$9 \cdot 10^3$	650 KB	101
10	512^3	512 MB	$27 \cdot 10^3$	1.9 MB	269
11	1024^3	4 GB	$96 \cdot 10^3$	6.6 MB	621
12	2048^3	32 GB	$413 \cdot 10^3$	29 MB	1129
13	4096^3	256 GB	$2 \cdot 10^6$	172 MB	1524
14	8192^3	2 TB	$15 \cdot 10^6$	1 GB	2048

Table 7.1: Comparison of memory requirements (approximate values) for 3D reconstruction of the barley depicted in Fig. 7.5 in a uniformely spaced voxel grid versus octree. Memory limits of a current consumer PC are reached for the regular grid already at a resolution of 1024^3, while an octree of depth 14 fits. This makes the octree a suitable data structure for 3D reconstruction of thin structures.

Performance Analysis

The memory requirements and resolution of the proposed method are compared to a standard volumetric approach using regular grids. A regular grid is a subdivision of a 3D volume into uniformely sized cuboids, also denoted as voxels. This yields a data volume with large amounts of empty voxels – in contrast to the octree with nodes of different sizes depending on the structure of the shape.

Tab. 7.1 shows a comparison of memory requirements for the octree data structure and the alternative representation using a regular grid. The values for the uniform grid were computed for each resolution while the values for the octree were measured experimentally for the example 3D model shown in Fig. 7.5. In each row of the table the actual size of a voxel is the same as the size of an octree node. Due to the connectivity of nodes the memory requirement for a single octree node is higher than the requirement for a single voxel, however the overall memory consumption is significantly reduced. For the uniform grid a resolution of 1024^3 reaches the limit of a current consumer PC with 4 GB RAM. The octree of depth 14 requires 1 GB, corresponding to a voxel volume of 8192^3. For a plant of 10 cm height, an octree node inside the visual hull covers a volume of $1.8 \cdot 10^{-6}$ mm^3, yielding a 2048 times higher resolution than a voxel of the regular grid fitting in the same memory, which covers a volume of 0.0037 mm^3. The experiment shows that the choice of data structure is crucial to make high-resolutions feasible for volumetric reconstructions.

7.5 High-Resolution 3D Shape Reconstruction for Plant Phenotyping

Accurate high-resolution 3D models are essential for a non-invasive analysis of phenotypic characteristics of plants. Leaf surface areas, fruit volumes and leaf inclination angles are typically of interest. This section shows how the presented method for 3D multi-view reconstruction method in octrees can be applied to phenotyping of plants. The thin structures typically occuring in the geometry of plants can be accurately reconstructed due to the high resolutions enabled by the use of octrees.

Plant phenotyping increasingly relies on precise 3D models of plants, demanding for automated and accurate reconstruction methods specialized for plant geometry. Applications include the determination of volume and surface dimensions, leaf quantification, and leaf inclination angles [118]. These applications share the benefit from accurate high-resolution 3D plant models. Since manual examination of phenotypic characteristics is usually time consuming and destructive, non-invasive and automated methods are needed for high-throughput applications and monitoring of specimen over time.

Full 3D models of plants allow for phenoypic analysis including the computation of volumes and surface areas or leaf inclination angles. Further analysis like monitoring of plant growth is possible since the plants are not destroyed during the process of reconstruction.

However, phenotyping is a major bottleneck in crop plant research [81], which strongly benefits of automated approaches especially when dealing with large datasets. A special importance lies on high-resolution reconstruction of plant shapes for a better comprehension of phenotypes [44]. Laser scanners are a capable tool for the aquisition of high-precision 3D point clouds of plants [97], however provide no volumetric and surface area information. Time-of-flight cameras and RGB-D sensors like the *Kinect* capture 3D information at a lower resolution. They are also used in agriculture however are known to be less robust to bright illumination than stereo vision [12, 78]. In the last years, image analysis has become a widely used technique for non-invasive methods for plant phenotyping [50]. Applications include the monitoring of growth rates which can be used as a measure for drought tolerance of wheat and barley [108], classification of leaves and stems [115], or computation of leaf inclination angles [19]. Volumetric information is an advantage when monitoring volumes and surface areas of plant canopies or fruits. An image-based interpretation tool for the estimation of the dimension of grapevine berries has been presented in [80]. In [129] an image based method for 4D reconstruction of plants based on optical flow is introduced. Another application is the determination of the leaf canopy area from images [91], an ecological indicator variable whose estimation usuallly is laborious [87].

7.5.1 Measuring Volumes and Surface Areas

The volume and surface area of a plant are fundamental indicators for growth analysis [117]. Volumetric models have the advantage that precise information on these features can be directly extracted from the shape.

The volume $\mathcal{V}(u)$ measured in voxels can be computed from the segmented surface u using equation (7.7). To obtain absolute measurements in cm^3, a reference measurement is necessary, for example the overall height h of the plant in cm, or in case of fixed cameras the baselines between the camera optical centers. The absolute volume $V(u)$ of the plant model can then be computed by a respective scaling of $\mathcal{V}(u)$, i.e. with $h^3/2^{3d}$, where d is the depth of the octree. If no reference measurement is given, the volume can be computed up to a constant scalar factor.

The surface area $A(u)$ corresponds to the boundary size of the reconstructed shape and can be computed from u with

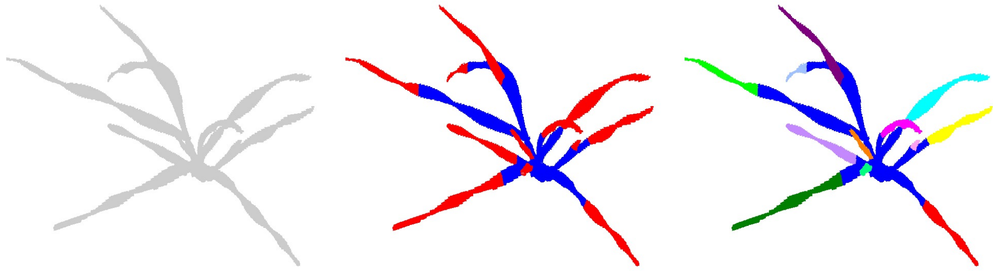
$$A(u) = \int_{\mathcal{H}} |\nabla u| \, \mathrm{d}x. \tag{7.14}$$

For the barley shown in Fig. 7.5 we measured a volume of $V(u) \approx 3.101$ cm^3 and a surface area of $A(u) \approx 106.1$ cm^2 for a plant height of 10 cm.

7.5.2 Quantification of Leaves

The total leaf number of a plant is an important trait used to monitor vegetative development. It can be used as an indicator to measure influences of drought [65] or to determine flowering times [104].

Full 3D models of plant shapes allow for automated quantification of leaves as the experiment in Fig. 7.7 shows for a barley. The reconstructed 3D model (Fig. 7.7 (a)) is segmented into two regions according to the eigenvalues of the second-moments tensor of the surface (Fig. 7.7 (b)). The



(a) 3D Model of a Barley

(b) Segmented Surface

(c) Connected Components

Figure 7.7: Segmentation of the 3D surface, based on the eigenvalues of the second-moments tensor. The connected components of the segmentation yield quantitative information like the number of leaves in the plant.

3D second-moments tensor [18] of a shape u is defined as

$$M(u) = \int_{\mathcal{H}} G_\sigma * \nabla u \nabla u^\top \, \mathrm{d}x \tag{7.15}$$

where G_σ is a gaussian convolution with standard deviation σ. The eigenvalues of M represent the distribution of gradient directions of the shape, and thus provide a robust classifier for a segmentation based on local geometric structures, see Fig. 7.8. Due to the high resolution of the 3D model the eigenvalues can be computed precisely. The connected components (Fig. 7.7 (c)) of the resulting segmentation allow for an automated quantification of leaves.

7.6 Conclusion

In this chapter a convex formulation for reconstruction of high-resolution volumetric 3D models from a set of RGB images was described. It was shown that the octree data structure is especially suitable for volumetric reconstruction of thin features. Moreover, it was shown that the choice of a suitable data structure is essential to make high-resolution 3D model reconstruction possible. Compared to standard data structures, like regular grids, up to 2000 times higher resolutions are feasible.

Thin structures typically occur in plant geometry. The reconstructed full 3D models allow for accurate phenotypic analysis of the geometric properties of plants including volume and surface areas or quantification of leaves. Possible future work includes a space-time reconstruction of plant growth. The

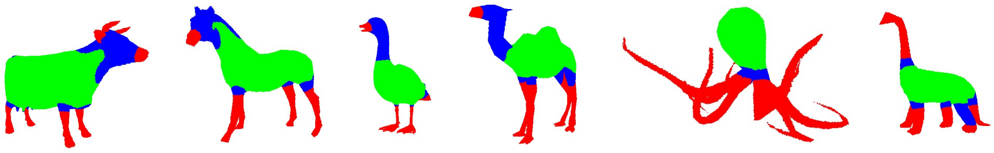


Figure 7.8: Unsupervised segmentation of surfaces based on the second moments tensor. The data terms were initialized using a *k-means* clustering algorithm.

non-invasiveness of the method allows for a monitoring of specimen over a time period.

Chapter 8

Conclusion

Starting from object identification in mobiles phones, movement tracking of game consoles to visual sensing in autonomous robots and self driving cars, not to mention the multitude of industrial and agricultural applications: Computer Vision methods have been established in many areas of every day life and are becoming ever more important. This thesis presented advances in several important fields of computer vision: image segmentation, object tracking, 3D stereo reconstruction for depth map estimation and full 3D multi-view reconstruction. The basic method applied in this thesis to all these fields is convex relaxation. The following briefly summarizes the main results and contributions of the thesis. Afterwards, an outlook to possible future work will be discussed.

8.1 Summary

Energy minimization has been established as one of the most successful basic techniques for many computer vision methods. This is due to the fact that they allow for an elegant mathematical description of the underlying computer vision problems. In addition, convex frameworks allow for global optimization, disposing of the need for finding good initial estimates first.

As a basis for the work in the further chapters, in Chapter 3 an experimental comparison of two of the most popular, but fundamentally different approaches to global optimization in computer vision was presented: discrete optimization with graph cuts was compared to continuous optimization with convex relaxation regarding run-time, memory consumption and accuracy. The experimental comparison revealed that graph cuts perform better with respect to run-times, while convex relaxation method yield better results with respect to memory consumption and accuracy. Based on these results, convex relaxation was chosen for the work presented in this thesis.

Image segmentation is an important problem in computer vision since it is a basis for many applications like object detection and recognition as well as medical image analysis. The task of image segmentation is easier to solve if the depicted objects are known a priori. An approach to apply a priori knowledge is to constrain the shape of the segmentation. Most approaches in this area rely on relatively strict shapes and are therefore only applicable to pre-learned reference shapes which allows for small deviations only. This thesis however presents a more generalized approach in a convex framework in Chapter 4. By applying constraints based on the central moments of a shape, basic properties of the desired shape can be enforced while also allowing for a certain deviation. This is especially important in the domain of medical imaging, where the object to segment may differ from the optimal shape. The presented method for image segmentation using convex moment constraints allows for visible and measurable improvements of segmentation results. On a quantitative evaluation on medical images it was shown that the segmentation error could be reduced from 12% to 0.35%. Efficient parallel implementations on the GPU allow for interactive applications. Furthermore, an extension of the method to object tracking in image sequences was shown.

Sensors that are able to directly generate color images with additional depth information (RGB-D) are a relatively new phenomenon. In Chapter 5 extensions of the moment constraints segmentation and object tracking method for 3D shape constraints in RGB-D data was shown. The scale-aware formulation allows for tracking an object's *absolute* dimensions rather than its *projected* dimensions in the image plane. This allows for robust object tracking with camera motion towards or away from the scene.

Despite their undisputed usefulness in certain well defined situations (especially in narrow indoor environments) RGB-D sensors are, due to their physical constraints, not especially suited for other situations, e.g., in outdoor environments. Stereo reconstruction methods applied to images taken with standard consumer cameras allow for relatively cheap and reliable methods to reconstruct the 3D geometry of a scene. Chapter 6 showed convex relaxation methods for stereo reconstruction. Due to the fact that image edges often correspond to object boundaries, an inclusion of this knowledge into the reconstruction process is of interest. Therefore, a convex method for edge-based disparity map reconstruction based on anisotropic regularization was presented. Experiments have shown that including image edges to stereo reconstruction can substantially improve reconstruction results.

A challenging real-world application of stereo reconstruction is the phenotyping of grapevine growth using images directly captured in the field from a mobile platform. The proposed stereo reconstruction method showed robust results for this application. Using this method, monitoring of grapevine growth for objective comparison of cultivars and estimation of fruit-to-leaf ratios was carried out with promising results.

While in high-throughput phenotyping of plants in the field, color and depth images can be sufficient, other applications require highly detailed full 3D reconstructions. In contrast to sparse reconstruction methods resulting in point clouds, dense reconstructions allow for direct deduction of shape properties like surface area and volume. Since explicit shape representations suffer from complex regridding problems during the optimization process, implicit representations allow for more elegant formulations. Chapter 7 presents a convex formulation of volumetric multi-view stereo reconstruction. The convex relaxation approach enables globally optimal solutions with a continuous representation.

The drawback of using an implicit representation is the high memory consumption, especially when highly detailed structures are to be reconstructed. Chapter 7 demonstrates that despite the implicit volumetric representation high-resolutions of up to $1.8 \cdot 10^{-6}$ mm^3 can be achieved due to the use of the memory-efficient octree data structure.

8.2 Future Work

While this thesis has shown advances in several fields of computer vision, many opportunities for extending the presented methods exist. In the following some of these opportunities are discussed.

Moment constraints are able to substantially improve image segmentation results with little user input. This thesis has shown some general forms of shape constraints for image segmentation, but it would be interesting to extend the presented method to other low-order properties of shapes, like planarity, convexity, thin structures, orientation, object's locations with respect to each other, etc. In particular, respective methods with continuous convex formulations are sparse. Furthermore, the method can be extended to the learning of class-specific covariance or ratio priors. By using soft constraints one could allow for classes to consist of objects that differ in the shape details, however share the same low-level properties. The respective shape constraints can help to improve segmentation results and object tracking. Another interesting research question is how the respective moment constraints can be learned from classes of reference shapes.

The presented scale-aware object tracking method is able to robustly track objects with 3D moment constraints. To allow for arbitrary movement of the tracked object, further research has to be done. An extension for different kinds of sensors, e.g. time of flight (ToF) cameras, would widen the applicability of the presented method. It would be interesting to investigate if it is possible to generalize the method to 4D space-time reconstruction with volume constraints and respective higher order constraints.

The stereo and segmentation for phenotyping of grapevine has shown that the presented method has a high potential for automated image analysis of large data bases. It can be generalized to other kinds of classes and other types of plants. Furthermore, the utilized algorithms can be fused into one, computing disparities and classifications simultaneously. This can enable high-throughput analysis of crop. The approach is furthermore generalizable to space-time reconstruction with temporal smoothness.

The functional lifting approach for stereo reconstruction used in this thesis converges very slow, due to the optimization of the disparity maps in higher dimensional space. Run-time improvements in form of efficient algorithms and implementations would be necessary to enable real-time applications.

Improving the maximal possible resolution of a volumetric space representation using subdivision schemes like octrees is possible only up to a certain point due to memory limitations. Especially for larger scenes like whole fields of plants or 3D reconstruction on a city scale it would be interesting how to overcome these limitations. One possible research direction could be the combination of recent developments from the field of massive point cloud management with the reconstruction methods. Another approach, which could be researched would be a combination of volumetric space representations with less memory consuming explicit representations.

Notation

$\Omega \subseteq \mathbb{R}^d$	Image domain of dimension d
$I : \Omega \to \mathbb{R}^b$	Input image with b color channels
x	Point, i.e. a pixel in an image or a voxel in a volume
Ω_i	ith region of a segmentation
$C_i = \partial\Omega_i$	Contour of region Ω_i
$C = \bigcup_{i=1}^{n} C_i$	Contour of a segmentation
$u : \Omega \to [0,1]$	Implicit contour representation
∇	Gradient operator mapping a scalar field to a vector field
$\lvert\nabla u\rvert$	Gradient norm mapping a vector field to a scalar field
div	Divergence operator mapping a vector field to a scalar field
Δ	Laplace operator
Du	Distributional derivative
$u_x := \frac{\partial u}{\partial x}$, $u_y := \frac{\partial u}{\partial y}$	Derivative in x and y direction, respectively
G_σ	Gaussian convolution with standard deviation σ
$I_\sigma : \Omega \to \mathbb{R}^b$	Input image smoothed with Gaussian convolution
$g : \Omega \to [0,1]$	Weighting function of a regularizer term
$f : \Omega \to \mathbb{R}$	Regional data term measuring point-wise data fidelity
$S \subseteq \mathbb{R}^d$	Hypersurface in $\mathbb{R}^d$
$BV(\Omega; \{0,1\})$	Set of binary functions of bounded variation on Ω
$\mathcal{B} = BV(\Omega; [0,1])$	Convex hull of $BV(\Omega; \{0,1\})$
p	Dual variable
$\tau, \tau_p, \tau_u \in \mathbb{R}$	Time step of a numerical optimization scheme
$\mathcal{N}(x)$	Neighborhood of a pixel x
$\mu \in \mathbb{R}^d$	Centroid of a shape
$x^\top$	Transpose of a vector x
$d : \Omega \to \mathbb{R}$	Depth map
$v : \Omega \to \mathbb{R}$	Disparity map
$\gamma \in [0, \gamma_{\max}]$	Disparity
$\rho(x, \gamma)$	Stereo matching cost for pixel x in image I_1 and $x + \gamma$ in I_2
Π_i	Projection matrix to image I_i

Appendix A

Camera Calibration

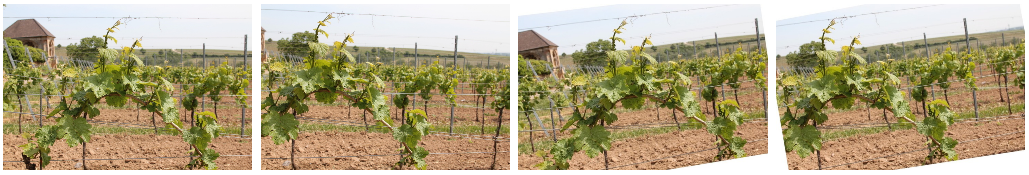
Intrinsic and extrinsic calibration of the used cameras are necessary preprocessing steps when 3D geometry is to be reconstructed. Intrinsic calibration aims at removing unwanted artifacts like lens distortion stemming from the inherent properties of the camera and can be applied to single images. Extrinsic calibration aims at estimating the camera position in 3D space, either the absolute position or the relative poses of multiple camera capturing positions to each other. Especially when dealing with two input images, rectification is usually the method of choice to transform the images to the stereo normal case. In the following, a short overview of intrinsic calibration, the image rectification process and computation of depth from disparity is given.

A.1 Lens Distortion and Intrinsic Calibration

Calibrating the intrinsic parameters of a camera is usually done with a chessboard or another previously measured pattern. It aims at inverting the distortion artifacts caused during the image capturing process stemming from the inherent properties of the camera. Estimated parameters usually include the focal length, a skew coefficient between the x and y axes, the principal point of the camera, and parameters of non-linear functions that model lens distortion. Except for the lens distortion, the parameters are constant for each pixel and can be represented in the intrinsic camera matrix

$$K = \begin{pmatrix} fm_x & s & o_x \\ 0 & fm_y & o_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{A.1}$$

where f is the focal length, m_x and m_y are scale factors to represent f in pixels, s is the skew coefficient between the x and y axes and o_x and o_y are



(a) Original Image Pair (b) Rectified Image Pair

Figure A.1: Example for Image Rectification. In the rectified image pair, epipolar lines are parallel, simplifying subsequent computations of disparities. Rectifications were computed using software of [56] and [99].

the coefficients of the principal point.

A.2 Pose Estimation with Epipolar Geometry

For the estimation of relative poses of camera positions to each other, it is necessary to identify pixels in both images depicting the same object point. Since relatively few data points are sufficient for a reliable reconstruction, most methods are based on sparse feature points, e.g. *SIFT* [99]. Matching the description vectors between these feature points yields the projection matrices $P = (R, T)$ with rotation matrix $R \in SO(3)$ and translation vector $T \in \mathbb{R}^3$. Methods for camera pose estimation include [56, 132]. Most methods try to reconstruct the fundamental matrix $F \in \mathbb{R}^{3\times 3}$ which maps a pixel x_1 in image I_1 to its corresponding pixel x_2 in image I_2. F is determined up to a scalar factor and satisfies the epipolar constraint:

$$x_2^\top F x_1 = 0. \tag{A.2}$$

A.3 Image Rectification

Image rectification is a pre-processing step for disparity map estimation. In rectified images, epipolar lines are parallel which simplifies the subsequent computation of disparity maps. Rectification implies a reprojection of both images, so that both projected images lie in the same plane and geometrical distortions are corrected. A rectified image pair is also denoted as *stereo normal case*. To rectify an image pair, the camera parameters are calibrated, i.e. intrinsic parameters and relative camera positions are estimated. With these parameters homographies are computed that facilitate the image transformation. The homography matrices $H_1, H_2 \in \mathbb{R}^{3\times 3}$ satisfy

$$x_2^\top \tilde{F} x_1 = 0, \text{ s.t. } \tilde{F} = H_2^\top F H_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{A.3}$$

which corresponds to a translation of 1 along the x axis and no rotation. Fig. A.1 shows an example of an input image pair and the corresponding rectified image pair, computed with software of [99] for the SIFT feature extraction and [56] for the rectification from point correspondences.

A.4 Depth from Disparity

Depth maps can be computed from disparity maps. A disparity map assigns to each pixel in a reference image I_1 the displacement v (also denoted by the *disparity*) of image locations of an object point seen in both images I_1 and I_2. The depth d is proportional to the inverse of the disparity v and can be computed by

$$d(x) = \frac{bf}{v(x)}. \tag{A.4}$$

Here f is the focal length of the camera and b is the baseline, i.e. the distance between the two camera capturing positions. Disparity maps give measurements in pixel units. However, if the camera parameters are known, absolute distances can be computed using (A.4). In particular, the baseline is proportional to the depth. If no reference measurement is given, depth maps can be computed up to a constant scalar factor, corresponding to the unknown baseline.

Publications

[1] S. Frintrop, M. Klodt, and E. Rome. A Real-time Visual Attention System Using Integral Images. In *5th International Conference on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.

[2] K. Herzog, R. Roscher, M. Wieland, M. Klodt, W. Förstner, H. Kuhlmann, D. Cremers, and R. Töpfer. Die Aufnahme von Stereobildern und Entwicklung automatisierter Verfahren zur Bildinterpretation für eine Hochdurchsatz-Phänotypisierung von Reben im Freiland. *Geilweilerhof aktuell*, 40(2):16–23, 2012.

[3] M. Klodt and D. Cremers. A Convex Framework for Image Segmentation with Moment Constraints. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[4] M. Klodt and D. Cremers. High-Resolution Plant Shape Measurements from Multi-View Stereo Reconstruction. In *ECCV Workshop on Computer Vision Problems in Plant Phenotyping*, Zürich, Switzerland, September 2014.

[5] M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. An Experimental Comparison of Discrete and Continuous Shape Optimization Methods. In *European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008.

[6] M. Klodt, F. Steinbruecker, and D. Cremers. Moment Constraints in Convex Optimization for Segmentation and Tracking. In *Advanced Topics in Computer Vision*. Springer, 2013.

[7] M. Klodt, J. Sturm, and D. Cremers. Scale-Aware Object Tracking with Convex Shape Constraints on RGB-D Images. In *German Conference on Pattern Recognition (GCPR)*, Saarbrücken, Germany, September 2013.

[8] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Propagated Photoconsistency and Convexity in Variational Multiview 3D Reconstruction. In

ICCV Workshop on Photometric Analysis for Computer Vision, Rio de Janeiro, Brazil, October 2007.

[9] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous Global Optimization in Multiview 3D Reconstruction. *International Journal of Computer Vision (IJCV)*, 84(1):80–96, August 2009.

[10] K. Kolev, M. Klodt, T. Brox, S. Esedoglu, and D. Cremers. Continuous global optimization in multiview 3D reconstruction. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2007.

[11] S. May, M. Klodt, E. Rome, and R. Breithaupt. GPU-accelerated Affordance Cueing based on Visual Attention. In *International Conference on Intelligent Robots and Systems (IROS)*, San Diego, California, 2007.

Remark on Prior Publication

Parts of this thesis have been published in [3, 4, 5, 6, 7, 8, 9, 10].

Bibliography

[12] S. M. Abbas and A. Muhammad. Outdoor rgb-d slam performance in slow mine detection. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. VDE, 2012.

[13] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1994.

[14] B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):106–118, 2006.

[15] J. Arnó, A. Escolà, J. Vallès, J. Llorens, R. Sanz, J. Masip, J. Palacín, and J. Rosell-Polo. Leaf area index estimation in vineyards using a ground-based lidar scanner. *Precision Agriculture*, 14(3):290–306, 2013.

[16] T. Bates, B. Grochalsky, and S. Nuske. Automating measurements of canopy and fruit to map crop load in commercial vineyards. *Research Focus: Cornell Viticulture and Enology*, 4:1–6, 2011.

[17] S. K. Behera, P. Srivastava, U. V. Pathre, and R. Tuli. An indirect method of estimating leaf area index in jatropha curcas l. using lai-2000 plant canopy analyzer. *Agricultural and Forest Meteorology*, 150(2):307–311, 2010.

[18] J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *IEEE First International Conference on Computer Vision (ICCV)*, pages 433–438, London, Great Britain, June 1987.

[19] B. Biskup, H. Scharr, U. Schurr, and U. Rascher. A stereo imaging system for measuring structural parameters of plant canopies. *Plant Cell Environ*, 30(10):1299–308, 2007.

[20] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[21] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference*, pages 1–11, 2011.

[22] 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.

[23] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. International Conference on Computer Vision*, pages 26–33, Nice, 2003.

[24] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 26(9):1124–1137, 2004.

[25] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. IEEE Conf. on Comp. Vision Patt. Recog. (CVPR'98)*, pages 648–655, Santa Barbara, California, 1998.

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

[27] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(11):1222–1239, 2001.

[28] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. *Lecture Notes in Statistics*, 37:28–47, 1986.

[29] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Fast Global Minimization of the Active Contour/Snake Model. *Journal of Mathematical Imaging and Vision*, 2007.

[30] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. IEEE Intl. Conf. on Comp. Vis.*, pages 694–699, Boston, USA, 1995.

[31] A. Chambolle. Total variation minimization and a class of binary MRF models. In *Int. Conf. on Energy Minimization Methods for Computer Vision and Pattern Recognition*, number 3757 in LNCS, pages 136–152. Springer, 2005.

[32] A. Chambolle, D. Cremers, and T. Pock. A convex approach for computing minimal partitions. *Communications on Pure and Applied Mathematics*, 2008.

[33] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[34] 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.

[35] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. Technical Report 54, UCLA, September 2004.

[36] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[37] T.F. Chan and L.A. Vese. A level set algorithm for minimizing the Mumford–Shah functional in image processing. In *IEEE Workshop on Variational and Level Set Methods*, pages 161–168, Vancouver, CA, 2001.

[38] 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(6):1161–1174, 2011.

[39] D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, 2006.

[40] A. Cutini, G. Matteucci, and G. S. Mugnozza. Estimation of leaf area index with the li-cor lai 2000 in deciduous forests. *Forest Ecology and Management*, 105(1-3):55–65, 1998.

[41] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 27(1-2):206–219, 2008.

[42] A. Delaunoy, E. Prados, P. Gargallo I Piracés, J.-P. Pons, and P. Sturm. Minimizing the Multi-view Stereo Reprojection Error for Triangular Surface Meshes. In *BMVC 2008 - British Machine Vision Conference*, pages 1–10, Leeds, September 2008. BMVA.

[43] A. Delong and Y. Boykov. A scalable graph-cut algorithm for n-d grids. In *Proc. International Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.

[44] S. Dhondt, N. Wuyts, and D. Inzé. Cell to whole-plant phenotyping: the best is yet to come. *Trends in Plant Science*, 18(8):433–444, 2013.

[45] M.-P. Diago, C. Correa, B. Millán, P. Barreiro, C. Valero, and J. Tardaguila. Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. *Sensors*, 12(12):16988–17006, 2012.

[46] E. A. Dinic. Algorithm for the solution of a problem of maximum flow in a network with power estimation. *Soviet Mathematics Doklady*, 11:1277–1280, 1970.

[47] J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19:248 – 264, 1972.

[48] P. Etyngier, F. Segonne, and R. Keriven. Shape priors using manifold learning techniques. In *Proc. International Conference on Computer Vision*, Rio de Janeiro, Oct 2007.

[49] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, March 1998.

[50] F. Fiorani and U. Schurr. Future scenarios for plant phenotyping. *Annual review of plant biology*, 64:267 – 291, 2013.

[51] L. Ford and D. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, New Jersey, 1962.

[52] A. Foulonneau, P. Charbonnier, and F. Heitz. Affine-invariant geometric shape priors for region-based active contours. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 28(8):1352–1357, 2006.

[53] D. Freedman and P. Drineas. Energy minimization via graph cuts: settling what is possible. In *Proc. International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 939–946, San Diego, USA, June 2005.

[54] P. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *In Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.

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

[56] A. Fusiello and L. Irsara. Quasi-euclidean epipolar rectification of uncalibrated images. *Machine Vision and Applications*, 22(4):663 – 670, 2011.

[57] P. Gargallo, E. Prados, and P. Sturm. Minimizing the Reprojection Error in Surface Reconstruction from Images. In *ICCV 2007 - 11th IEEE International Conference on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil, October 2007. IEEE Computer Society.

[58] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 6(6):721–741, 1984.

[59] A. Goldberg and R. Tarjan. A new approach to the maximum flow problem. *Journal of the ACM*, 35(4):921–940, 1988.

[60] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45:783–797, 1998.

[61] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 459–464, 1999.

[62] L. Gorelick, F. R. Schmidt, Y. Boykov, A. Delong, and A. Ward. Segmentation with non-linear regional constraints via line-search cuts. In *European Conference on Computer Vision (ECCV)*, Florence, Italy, Oct 2012.

[63] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis. Convexity shape prior for segmentation. In *European Conference on Computer Vision*, September 2014.

[64] G. Graber, T. Pock, and H. Bischof. Online 3d reconstruction using convex optimization. In *1st Workshop on Live Dense Reconstruction From Moving Cameras, ICCV 2011*, 2011.

[65] C. Granier, L. Aguirrezabal, K. Chenu, S.J. Cookson, M. Dauzat, P. Hamard, J. Thioux, G. Rolland, S. Bouchier-Combaud, and A. Lebaudy. Phenopsis, an automated platform for reproducible phenotyping of plant responses to soil water deficit in arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit. *New Phytologist*, 169:623–635, 2006.

[66] M. Grasmair and F. Lenzen. Anisotropic total variation filtering. *Applied Mathematics and Optimization*, 62(3):323–339, 2010.

[67] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum *a posteriori* estimation for binary images. *J. Roy. Statist. Soc., Ser. B.*, 51(2):271–279, 1989.

[68] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes.* Springer, New York, 1991.

[69] K. Herzog, R. Roscher, M. Wieland, A. Kicherer, T. Läbe, W. Förstner, H. Kuhlmann, and R. Töpfer. Initial steps for high-throughput phenotyping in vineyards. *Vitis*, 53(1):1–8, 2014.

[70] H. Hirschmüller. Evaluation of cost functions for stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[71] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

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

[73] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 23:253–258, 1925.

[74] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(10):1075–1088, 2001.

[75] L. F. Johnson and L. L. Pierce. Indirect measurement of leaf area index in california north coast vineyards. *HortScience*, 39(2):236–238, 2004.

[76] I. Jonckheere, S. Fleck, K. Nackaerts, B. Muys, P. Coppin, M. Weiss, and F. Baret. Review of methods for in situ leaf area index determination: Part i. theories, sensors and hemispherical photography. *Agricultural and Forest Meteorology*, 121(1âĂŞ2):19–35, 2004.

[77] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *IJCV*, 1(4):321–331, jan 1988.

[78] W. Kazmi, S. Foix, G. Alenyà, and H. J. Andersen. Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88(0):128 – 146, 2014.

[79] S. Kichenassamy, A. Kumar, P. J. Olver, A. Tannenbaum, and A. J. Yezzi. Gradient flows and geometric active contour models. In *Proc. International Conference on Computer Vision*, pages 810–815, 1995.

[80] A. Kicherer, R. Roscher, K. Herzog, S. Simon, W. Förstner, and R. Töpfer. Bat (berry analysis tool): A high-throughput image interpretation tool to acquire the number, diameter, and volume of grapevine berries. *Vitis*, 52(3):129–135, 2013.

[81] B. Kilian and A. Graner. Ngs technologies for analyzing germplasm diversity in genebanks. *Brief Funct Genomics*, 11(1):38–50, Jan 2012.

[82] K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008.

[83] K. Kolev, T. Pock, and D. Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In *European Conference on Computer Vision (ECCV)*, Heraklion, Greece, September 2010.

[84] V. Kolmogorov and Y. Boykov. What metrics can be approximated by Geo Cuts or global optimization of length/area and flux. In *Proc. International Conference on Computer Vision*, Beijing, 2005.

[85] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in vision. In *Proc. International Conference on Computer Vision*, Rio de Janeiro, Brasil, 2007.

[86] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24(5):657–673, 2004.

[87] L. Korhonen and J. Heikkinen. Automated Analysis of in Situ Canopy Images for the Estimation of Forest Canopy Cover. *Forest Science*, 55(4):323–334, August 2009.

[88] G. Kuschk and D. Cremers. Fast and accurate large-scale stereo reconstruction using variational methods. In *ICCV Workshop on Big Data in 3D Computer Vision*, Sydney, Australia, December 2013.

[89] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pages 239–253, Berlin, Heidelberg, 2010. Springer-Verlag.

[90] A. Ladikos, S. Benhimane, and N. Navab. Multi-view reconstruction using narrow-band graph-cuts and surface normal optimization. In *BMVC*, pages 1–10, 2008.

[91] R. N. Lati, S. Filin, and H. Eizenberg. Robust methods for measurement of leaf-cover area and biomass from image data. *Weed Science*, 59(2):276–284, April-June 2011.

[92] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):150–162, February 1994.

[93] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *ICCV*, pages 646–653. IEEE, 2009.

[94] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *LNCS*, pages 150–162. Springer, 2009.

[95] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *Proc. International Conference on Computer Vision*, Kyoto, Japan, 2009.

[96] Y. Lim, K. Jung, and P. Kohli. Constrained discrete optimization via dual space search. In *NIPS Workshop on Discrete Optimization in Machine Learning (DISCML)*, 2011.

[97] G. Louarn, S. Carré, F. Boudon, A. Eprinchard, and D. Combes. Characterization of whole plant leaf area properties using laser scanner point

clouds. In *Fourth International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications*, Shanghai, Chine, 2012.

[98] G. Louarn, J. Lecoeur, and E. Lebon. A three-dimensional statistical reconstruction model of grapevine (vitis vinifera) simulating canopy structure variability within and between cultivar/training system pairs. *Annals of Botany*, 101(8):1167–1184, 2008.

[99] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[100] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

[101] H. Mabrouk and H. Sinoquet. Indices of light microclimate and canopy structure of grapevines determined by 3d digitising and image analysis, and their relationship to grape quality. *Australian Journal of Grape and Wine Research*, 4(1):2–13, 1998.

[102] F. Mazzetto, A. Calcante, A. Mena, and A. Vercesi. Integration of optical and analogue sensors for monitoring canopy health and vigour in precision viticulture. *Precision Agriculture*, 11(6):636–649, 2010.

[103] D. Meagher. *Octree Encoding: a New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer.* Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory, 1980.

[104] B. Mendez-Vigo, M. de Andres, M. Ramiro, J. Martinez-Zapater, and C. Alonso-Blanco. Temporal analysis of natural variation for the rate of leaf production and its relationship with flowering initiation in arabidopsis thaliana. *Journal of Experimental Botany*, 61(6):1611–23, 2010.

[105] G. E. Meyer. Machine vision identification of plants. *Recent Trends for Enhancing the Diversity and Quality of Soybean Products*, 2011.

[106] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of r^n. *J. Optimization Theory and Applications*, 50(1):195–200, 1986.

[107] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.

[108] R. Munns, R. A. James, X. R. R. Sirault, R. T. Furbank, and H. G. Jones. New phenotyping methods for screening wheat and barley for beneficial responses to water deficit. *Journal of Experimental Botany*, 61(13):3499–3507, August 2010.

[109] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *Workshop on Consumer Depth Cameras for Computer Vision (CDC4CV)*, pages 1147–1153, 2011.

[110] C. Nieuwenhuis and D. Cremers. Spatially varying color distributions for interactive multi-label segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 35(5):1234–1247, 2013.

[111] C. Nieuwenhuis, E. Strekalovskiy, and D. Cremers. Proportion priors for image sequence segmentation. In *Proc. International Conference on Computer Vision*, Sydney, Australia, December 2013.

[112] C. Nieuwenhuis, E. Toeppe, and D. Cremers. A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model. *International Journal of Computer Vision*, 2013.

[113] S. J. Osher and J. A. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. of Comp. Phys.*, 79:12–49, 1988.

[114] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York, 4th edition edition, 2002.

[115] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp. A novel mesh processing based technique for 3d plant analysis. *BMC Plant Biology*, 12(1):63, 2012.

[116] S. Paulus, J. Dupuis, A.-K. Mahlein, and H. Kuhlmann. Surface feature based classification of plant organs from 3d laserscanned point clouds for plant phenotyping. *BMC Bioinformatics*, 14:238, 2013.

[117] S. Paulus, H. Schumann, H. Kuhlmann, and J. Leon. High-precision laser scanning system for capturing 3d plant architecture and analysing growth of cereal plants. *Biosystems Engineering*, 121:1–11, May 2014.

[118] J. Pisek, O. Sonnentag, A. D. Richardson, and M. Mottus. Is the spherical leaf inclination angle distribution a valid assumption for temperate

and boreal broadleaf tree species? *Agricultural and Forest Meteorology*, 169(0):186 – 194, 2013.

[119] 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*, Kyoto, Japan, 2009.

[120] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *Proc. European Conference on Computer Vision*, Marseille, France, October 2008.

[121] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the Limits of Stereo Using Variational Stereo Estimation. In *IEEE Intelligent Vehicles Symposium*, 2012.

[122] R. Roscher, K. Herzog, A. Kunkel, A. Kicherer, R. Töpfer, and W. Förstner. Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. *Computers and Electronics in Agriculture*, 100(0):148–158, 2014.

[123] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[124] L. Rudin, S. Osher, and C. Fatemi. Nonlinear total variation based noise removal algorithms. *physicaD*, 60:259–268, 1992.

[125] J. Santner, T. Pock, and H. Bischof. Interactive multi-label segmentation. In *Proceedings 10th Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand*, November 2010.

[126] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, April 2002.

[127] C. Schmaltz, P. Peter, M. Mainberger, F. Ebel, J. Weickert, and A. Bruhn. Understanding, optimising, and extending data compression with anisotropic diffusion. *International Journal of Computer Vision*, 108(3):222–240, 2014.

[128] T. Schoenemann and D. Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[129] T. Schuchert and H. Scharr. Estimation of 3d object structure, motion and rotation based on 4d affine optical flow using a multi-camera array. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 596–609, Berlin, Heidelberg, 2010. Springer-Verlag.

[130] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 22(8):888–905, 2000.

[131] R. E. Smart, J. K. Dick, I. M. Gravett, and B. M. Fisher. Canopy management to improve grape yield and wine quality-principles and practices. *South African Journal for Enolgy and Viticulture*, 11(1):3–17, 1990.

[132] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press.

[133] F. Steinbruecker, C. Kerl, J. Sturm, and D. Cremers. Large-scale multi-resolution surface reconstruction from rgb-d sequences. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, 2013.

[134] G. Strang. Maximal flow throug a domain. *Mathematical programming*, 26(2):123–143, 1983.

[135] J. Stühmer, P. Schröder, and D. Cremers. Tree shape priors with connectivity constraints using convex relaxation on general graphs. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[136] C.J. Taylor and A. Cowley. Fast scene analysis using image and range data. In *Proc. of the Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[137] E. Toeppe, C. Nieuwenhuis, and D. Cremers. Volume constraints for single view reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, 2013.

[138] R. Töpfer, L. Hausmann, M. Harst, E. Maul, E. Zyprian, and R. Eibach. *New horizons for grapevine breeding*, volume 5, pages 79–100. Global Science Books, , 2011.

[139] M. Unger, T. Mauthner, T. Pock, and H. Bischof. Tracking as segmentation of spatial-temporal volumes by anisotropic weighted tv. In *7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 5681 of *LNCS*, pages 193–206, Bonn, Germany, 2009. Springer.

[140] M. Unger, T. Pock, and H. Bischof. Continuous globally optimal image segmentation with local constraints. In *Computer Vision Winter Workshop 2008*, Moravske Toplice, Slovenija, February 2008.

[141] M. Unger, T. Pock, D. Cremers, and H. Bischof. Tvseg - interactive total variation based image segmentation. In *British Machine Vision Conference (BMVC)*, Leeds, UK, September 2008.

[142] M. Unger, M. Werlberger, T. Pock, and H. Bischof. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[143] O. Veksler. Star shape prior for graph-cut image segmentation. In *Proc. European Conference on Computer Vision*, pages 454–467, 2008.

[144] G. Vogiatzis, P. Torr, and R. Cippola. Multi-view stereo via volumetric graph-cuts. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 391–399, 2005.

[145] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009.

[146] A. J. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal of Computer Vision*, 53(1):31–43, June 2003.

[147] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, Stockholm, Sweden, May 1994.

[148] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision, Modeling and Visualization Workshop VMV 2008*, October 2008.