Kerstin Hammernik, BSc

Convex Framework for 2D & 3D Image Segmentation Using Shape Constraints

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieurin

Individual master's degree programme
Biomedical Imaging

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Thomas Pock
Institute for Computer Graphics and Vision

Advisor

Dipl.-Ing. Dr.techn. Martin Urschler
Ludwig Boltzmann Institute for Clinical Forensic Imaging

Graz, Austria, January 2015

To my grandmother.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort	Datum	Unterschrift

Abstract

Image segmentation is one among many difficult tasks in computer vision. Due to limitations of image modalities that lead to noise or weak boundaries, object occlusion or intensity inhomogeneities, it is beneficial to incorporate shape prior knowledge to obtain more robust segmentation results. In this thesis, we explore different kinds of shape constraints for interactive medical image segmentation embedded in a convex variational framework and show how we can decrease user interaction significantly.

We introduce two types of shape constraints. First, we show how we can describe shapes globally by means of star prior and moment constraints. While the first constraint ensures one-connected star convex objects, the latter constraint is responsible that first-order moments such as volume or center of gravity are fulfilled. Combining these constraints results in a powerful tool which can be applied to a variety of applications in 2D and 3D. We show the application to sinus floor augmentation segmentation and compare the results to an expert's segmentation.

The second type of constraints that we consider in this thesis are model specific shape constraints. While simple objects can be described by means of global shape constraints, more complex objects need a specific model description which requires prior knowledge from training data. Vertebrae segmentation is suitable to evaluate model specific shape constraints in this framework due to the complex substructures of vertebrae. A publicly available database allows for shape model estimation and provides experts' segmentations for quantitative evaluation. We achieve promising results on this database which are comparable to literature.

All proposed shape constraints can be specified by a single point or an ellipsoid. To interact with volumetric data, we provide a user-friendly Graphical User Interface (GUI). We exploit the high parallelization potential of our variational framework and implement the algorithms on the Graphics Processing Unit (GPU) using NVIDIA® CUDA.

Kurzfassung

Segmentierung von Bildern ist ein komplexer Bereich in der Bildverarbeitung. Aufgrund technischer Limitierungen bildgebender Verfahren, die in Bildrauschen und schwachen Kanten resultieren, sowie Verdeckung von Objekten und Belichtungsinhomogenitäten, ist es für eine robuste Segmentierung notwendig, Vorwissen über die Form des zu segmentierenden Objektes miteinzubeziehen. In dieser Arbeit werden zwei Möglichkeiten analysiert, um die Form eines Objekts in einem interaktiven Segmentierungsframework, das auf Variationsrechnung und konvexer Optimierung basiert, zu berücksichtigen. Ferner wird gezeigt, wie mit diesen Methoden die Benutzerinteraktion verringert werden kann.

Die erste Möglichkeit besteht darin, die Form eines Objektes global zu beschreiben. Ein Beispiel dafür ist der Star Prior, der zur Segmentierung sternkonvexer sowie zusammenhängender Objekte dient. Des Weiteren werden Momente betrachtet. Momente beschränken das Volumen oder den Schwerpunkt des Objektes. Eine Kombination der genannten globalen Beschreibungen der Form liefern vielversprechende Ergebnisse sowohl in 2D als auch in 3D. Eine medizinische Anwendung zur Quantifizierung von Knochenmaterial bei Sinuslift wird gezeigt und mit der Segmentierung eines Experten verglichen.

Während die Form einfacher Objekte global eingeschränkt werden kann, benötigen komplizierte Objekte ein detaillierteres Modell, das anhand von Trainingsdaten erstellt wird. Segmentierung von Wirbel eignet sich aufgrund deren komplexen Substruktur um dieses Modell zu untersuchen. Trainingsdaten einer öffentlich zugänglichen Datenbank erlauben es, ein spezifisches Modell zu erstellen und die Ergebnisse quantitativ mit Referenzsegmentierungen zu vergleichen. Es werden vielversprechende Ergebnisse erzielt, die mit Ergebnissen aus der Literatur vergleichbar sind.

Alle vorgestellten Modelle beruhen auf intuitiver Benutzerinteraktion in Form von Punkten oder eines Ellipsoides. Um mit volumetrischen Daten zu interagieren, wurde eine graphische Benutzeroberfläche entwickelt. Des Weiteren wird die Parallelisierbarkeit des Frameworks ausgenützt und die Algorithmen auf der Grafikkarte mit NVIDIA® CUDA implementiert.

Acknowledgments

Firstly, I would like to thank my supervisor Thomas Pock for his excellent guidance and support during my thesis. Thomas Pock always gave useful feedback and provided valuable answers to my questions. Secondly, I would like to express my gratitude to Martin Urschler, my thesis and my master's project advisor, for his untiring dedication. Martin Urschler always had time to answer my questions, have discussions and shared his expert knowledge in GPU programming.

Next, I would like to thank the members of the Institute for Computer Graphics and Vision not only for insightful discussions and useful input for my thesis but also for their ongoing support in and outside of academia . I'm particularly grateful to the Medical Image Processing group, namely Darko Stern and Thomas Ebner who I shared an office with.

Parts of this work were done in collaboration with the Department of Dentistry and Maxillofacial Surgery at the Medical University of Graz. I would like to thank Dr. med. dent. Susanne Vogl for her support in this project.

Furthermore, I would like to express my gratitude to all my friends, I appreciate the care and understanding they have shown me during this period. I'm especially grateful to Andreas Lesch, who convinced me to start an individual master's program, and for his collaboration in several lectures and project work during the last few years.

Moreover, I would like to thank my family for their love, support and that they gave me the opportunity to follow my passion.

Finally, I'm grateful to Michael for his love, support and patience.

Contents

List of Figures

List of Tables

List of Symbols

$\langle x, y\rangle$	Scalar product
∇	Nabla operator (discrete)
div	Divergence operator (discrete)
$f^*(\cdot)$	Convex conjugate function
$\lVert f\rVert_{L^p}$	L^p norm, defined as $\left(\int_\Omega \lvert f(x)\rvert^p \,\mathrm{d}\mu(x)\right)^{\frac{1}{p}}$
$\lVert x\rVert_{l_p}$, $\lVert x\rVert_p$	Discrete l_p norm, defined as $\lVert x\rVert_p = \left(\sum_{i=1}^{n} \lvert x_i\rvert^p\right)^{\frac{1}{p}}$
$\lVert x\rVert_{p,q}$	Inner l_p norm, outer l_q norm defined as $\left[\sum_{i}^{m}\left(\sum_{j}^{n}\lvert x_{ij}\rvert^p\right)^{q/p}\right]^{1/q}$
$\lVert\nabla x\rVert_{TV}$	Total Variation semi-norm, $\lVert\nabla x\rVert_{2,1}$
$\lVert\nabla_g x\rVert_{TV}$	g-Weighted Total Variation semi-norm, $\lVert\nabla_g x\rVert_{2,1}$
$\lVert\cdot\rVert$	l_2 norm
proj ()	Projection operator
prox ()	Proximal operator
Ω	Image domain
$\mathcal{L}(\cdot)$	Lagrangian function
$I_C(\cdot)$	Indicator function of the convex set C
δ^+	Forward differences
δ^-	Backward differences
$\mathbf{I}$	Identity matrix
$\mathbf{1}$	Vector of ones
L	Lipschitz constant
τ	Primal step size
$\mathbf{T}$	Diagonal matrix holding dimension-dependent primal step sizes τ_j
σ	Dual step size
$\mathbf{\Sigma}$	Diagonal matrix holding dimension-dependent dual step sizes σ_i

1

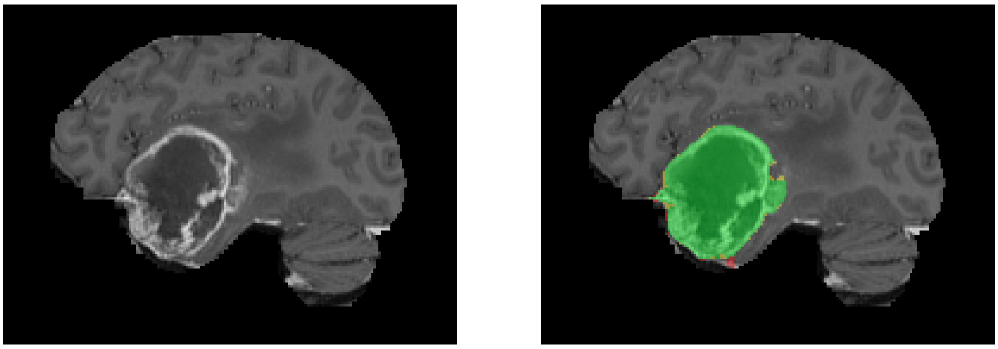
Introduction

Contents

1.1 Motivation

The objective of image segmentation is to subdivide a given input image into a set of meaningful, non-overlapping regions that share the same features such as intensity, color or texture. This is one of the most challenging problems in computer vision due to present noise, weak boundaries, object occlusion and intensity inhomogeneities because of non-uniform illumination. Image segmentation is crucial to a large number of applications in computer vision such as object detection, object recognition, tracking and scene understanding. Another essential field of application is medical image segmentation. It provides means to study and assess anatomical structures and pathologies in a non-invasive way. Medical image segmentation is a key part of computer-aided diagnosis, as it supports radiologists to monitor progress of diseases and effects of therapies and tissue quantification. Furthermore, medical image segmentation helps radiologists to detect, classify and visualize pathologies such as tumors. An example for brain tumor segmentation is illustrated in Fig. 1.1.

Up to now, medical image segmentation is often carried out manually. Manual segmentations done by experts are important reference or ground truth segmentations. However, the quality of the segmentations can only be assessed subjectively. Experts often do not



Segmentation Input image

Figure 1.1: Example for image segmentation of a brain tumor core. The image is taken from the BRATS data set [47].

agree about the correct segmentations (inter-observer variability). Furthermore, the segmentations of the same object carried out by one expert at different times do not overlap completely either (intra-observer variability). This indicates that the task of medical image segmentation is ambiguous.

While manual segmentation is too time consuming and the amount of image data is increasing, automatic segmentation methods which do not require any user interaction are desired to increase segmentation speed. Nevertheless, automatic methods are often not accurate and reliable enough. The disadvantages of manual and fully automatic methods give rise to the development of semi-automatic segmentation methods.

Semi-automatic image segmentation methods combine the advantages of manual and fully automatic segmentation methods, i.e., expert knowledge and powerful computations to increase segmentation accuracy. Expert or user knowledge can be incorporated in two different ways. One possibility is to offer a fast correction ability to refine segmentations arising from automatic methods. The second possibility is to use *prior knowledge* of the user to guide a segmentation algorithm. Prior knowledge can be incorporated on the one hand by means of user scribbles defining object and background and on the other hand by providing information about the object's shape.

Shape information is essential for accurate and robust segmentation. The definition of shape ranges from simple geometric shapes up to highly sophisticated shape models. This complex topic gives rise to explore different kinds of prior knowledge in terms of shape constraints and to embed shape information in an image segmentation framework.

1.2 Image Segmentation and Variational Methods

Image segmentation is a complicated and non-trivial task in computer vision. A formal definition is given as follows [33]:

- $\bigcup_{i=1}^{n} I_i = I$
- I_i is a connected set $\forall i$
- $I_i \cap I_j = \varnothing, \forall i, j, \; i \neq j$

where I is the input image and n is the number of disjunct subregions. These conditions implicate that every subregion is a connected region. The union over all subregions denote that every pixel is assigned to one subregion or label. In practice, we deal either with binary, i.e., object and background label, often used in interactive segmentation approaches, or multi-label segmentation problems.

We have mentioned that image segmentation is a very ambiguous task. Obtaining a reliable segmentation highly depends on the underlying image data, application and problem formulation. Many low-level segmentation algorithms such as thresholding, region growing and edge detection [33, 63] are not suitable for this task. During the last decades, high-level segmentation approaches have been studied extensively to achieve more robust segmentation results. Well known sophisticated algorithms are graph-cut segmentation [10], mean shift segmentation [20] and variational methods. Prominent variational methods for image segmentation are deformable contour segmentation [14, 41, 71], level-sets [60], continuous convex optimization methods and the Mumford-Shah functional [48]. The Mumford-Shah functional is one of the most studied variational problems in image segmentation. It approximates a given input image with piecewise smooth functions. The aim is to subdivide an image into a number of disjunct, untextured regions with smooth boundaries resulting in a cartoon-like representation of the input image. Chan and Vese [19] solved the Mumford-Shah functional using the level-set method, Pock [53] proposed a convex relaxation method for its approximate solution. During the last years, continuous convex optimization methods have become popular in several image processing tasks. Before we discuss the differences and advantages to other segmentation algorithms, we first introduce variational problems in general.

The aim of variational methods is to estimate an extremum of a given energy functional. A *functional* defines the mapping from a vector valued input function to a scalar value. It can be defined as the integral over a given domain, i.e., the region where the input function is defined. Changing the input function by an increment leads to a *variation* of the functional. Hence, variations are the same for functionals as derivatives are for functions. Finding the extremum of a functional implies that the functional has to be stationary, hence, the first variation is zero. In other words, we want to vary over all reasonable functions to find functions that give an extremal value for the functional. Solutions can be obtained using the discretized Euler-Lagrange equation combined with gradient descent methods to solve the respective partial differential equations numerically. Using convex functionals provide means for a globally optimal solution.

While level-set methods can easily get stuck in local minima, continuous convex optimization methods yield a globally optimal solution independent of initialization. Furthermore, continuous convex optimization methods do not suffer from metrication errors such as graph cuts. Another advantage is that discretization is only necessary when the energy functional is set up. Continuous, convex energy minimization methods can be applied to a variety of computer vision problems and thus constitute a powerful general framework. They offer high parallelization capabilities using Graphics Processing Units (GPUs). Additionally, this framework allows for an easy extension to incorporate prior knowledge by means of additional constraint terms. This is beneficial in order to restrict the space of possible solutions.

1.3 Prior Knowledge in Image Segmentation

Low-level features such as intensity values, texture and edges often produce poor segmentation results. This makes it necessary to include prior knowledge about objects of interest. In interactive segmentation, we use experts' knowledge to make restrictions on the objects' shape. Adams and Bischof [1] were one of the first who used foreground and background scribbles in interactive segmentation approaches. This idea has been successfully incorporated into the graph cut framework [5, 7, 8, 56] and continuous convex optimization approaches [51, 59, 66, 70]. The latter used foreground and background scribbles also to build intensity distributions which are used to predict whether a pixel is likely to be foreground or background, respectively. Besides prior user knowledge, it is essential to consider prior shape knowledge.

Shape is the geometric information that remains after removing the impact of translation, rotation and scaling [42]. We focus on two different kinds of shape representation, a global shape description and a model specific shape representation. Global shape knowledge views objects in a more general way and can be applied to a variety of applications, when the objects' shape fits the global shape constraints. Examples for global shape constraints are connectivity constraint [68], elliptical shape prior [62], constraints on image moments [44] and star convexity [35, 67] which require user initialization.

More complex objects cannot be described by means of simple geometric shapes and make it necessary to use application specific shape knowledge to increase the performance of segmentation algorithms. Specific shapes can be modeled in terms of a mean shape and statistical variations in shape. Nevertheless, building these shape models requires a set of training shapes. One possibility is to represent a shape instance in terms of a finite set of point correspondences (landmarks) [23]. All training shapes have to be annotated at the same landmarks. Afterwards, the shapes are aligned using the Procrustes algorithm [34]. Each single shape is transformed to the average shape such that the sum of squared distances of the point correspondences are minimized. Thus, similarity transformations, i.e., translation, rotation and scaling are taken into account. After aligning each shape, a new mean shape is defined and the algorithm is repeated until convergence. Statistical Shape Models (SSMs) are expressed by means of Principal Component Analysis (PCA) which models the variations in shape by using a mean shape in addition with eigenvectors describing the variations. Any shape can be approximated using the mean shape and a weighted combination of eigenvectors. This is known as Active Shape Model (ASM) [23]. The Active Appearance Model [22] is an extension of the ASM which additionally takes texture information into account. However, it is crucial to get correct landmark positions. While manual labeling is again time consuming, automatic methods are not accurate enough to estimate anatomical landmarks. Leventon et al. [46], Rousson et al. [57] and Cremers et al. [25] proposed to use signed distance functions calculated on the aligned binary training shapes as input for PCA. The advantage compared to point correspondences is to allow for slight misalignments of the shape. Cremers et al. [26] described how to use shape priors in variational image segmentation.

It has been shown that shape prior knowledge is essential to segment e.g. occluded objects [24]. In medical image segmentation, prior shape knowledge is a hot topic due to limitations of different medical imaging modalities and complex appearance of anatomical structures including pathological structures and biological variations requiring experts' knowledge.

1.4 Medical Image Segmentation and its Limitations

Medical imaging plays a vital role to study and assess anatomical structures in a non-invasive way. Combined with medical image segmentation, it gives not only insight into anatomical structures, but it offers also the possibility for computer-aided diagnosis of pathologies, therapy planning and quantification. However, biological variations and pathological anatomical structures as well as limitations of different image modalities influence the accuracy of segmentation algorithms. In the following, we give a short overview of basic terms that are necessary to understand medical images and commonly used medical image modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET) and Ultrasound (US) along with their limitations for medical image segmentation.

Medical images represent a physical domain (Field of View (FoV)) of an object as discrete pixels (2D) or voxels (3D). Voxel intensities have a quantitative or qualitative meaning depending on the image modality. The image spacing depicts the physical distance between voxels and defines the resolution of the image. The lower the image spacing is, the higher is the resolution. In general, we deal with anisotropic voxels. This means that the in-plane resolution is high compared to the axial resolution, i.e., slice thickness. Image resolution has a great impact on the visualization of image edges. The bigger a voxel is, the more likely it is that several tissues share the same voxel. This results in blurred or weak image edges and is known as partial volume effect which could be avoided by using smaller image spacings. However, the image resolution is limited by the underlying image modality and its Signal-to-Noise Ratio (SNR). Other problems that influence segmentation accuracy are noise and low contrast images as well as modality-dependent artifacts.

Computed Tomography uses X-rays sent out from different projection angles to image a tomography or slice of an object. Detector elements capture the reduced amount of radiation due to attenuation effects such as absorption and scattering. Back-projection algorithms are used to reconstruct image slices and estimate an attenuation coefficient for each voxel. The voxel intensities of CT images have a quantitative meaning and are expressed as Hounsfield Units (HU) which relates the attenuation coefficient to water, resulting in a scale that ranges from -1024 HU (air) to more than 1200 HU (dense bone tissue) in practice. A low resolution in CT images can mix up HU from different tissues, hence, it makes it more difficult to obtain an accurate segmentation. Nevertheless, a higher resolution requires an increase of applied radiation dose to achieve the same SNR. Common imaging artifacts in CT that influences segmentation quality are streaking artifacts due to beam hardening arising at high attenuation tissue such as metal implants or dense bone.

Magnetic Resonance Imaging does not use any ionizing radiation and provides excellent soft tissue contrast that cannot be attained by any other image modality. It is based on a strong magnetic field usually between 1.5 T - 7 T. In most medical applications, MRI focuses on hydrogen atoms that precess like gyroscopes in the static magnetic field. These protons can be excited using resonant Radio Frequency (RF) pulses and then induce a signal in a receiver coil. To obtain an image, the signal has to be encoded which is performed by varying the magnetic field by means of gradient fields. In contrast to CT, voxels do not usually have a quantitative meaning as they describe the signal intensity due to excited protons. If the resolution is increased, it results in signal loss. Common problems influencing segmentation algorithms are motion artifacts as well as intensity inhomogeneities due to inhomogeneous excitation and coil sensitivities.

An example for functional imaging is PET which measures the concentration of radionuclide (tracers) inside the body. The used radionuclides are positron emitters and characterized by a short half-life. When an emitted positron collides with an electron, they annihilate themselves and two gamma rays are produced carrying the energy of the destroyed positron and electron. The two gamma rays are emitted in opposite direction and are detected by two opposing gamma cameras simultaneously. This is called coincidence detection. As in CT, several projections are obtained which are necessary for image reconstruction. The image resolution of PET image is much lower compared to the other presented methods due to the low number of counted gamma rays and the geometry of detector elements of gamma cameras. This makes it difficult to depict sharp edges in the resulting images.

Ultrasound is a fast and comparably cheap method to examine visceral organs, female pelvic organs and fetuses without using harmful radiation. Sound waves above the hearing range are transmitted into the body using a transducer. The transmitted sound waves interact with tissue, i.e., the waves can be scattered, transmitted or reflected. Reflected sound waves generate an echo which is recorded. For imaging, an array of transducers is used and recorded echoes are converted to brightness values. Other modes exist to examine for example blood flow. However, it needs a lot of experience to assess ultrasound images due to high noise level, low contrast and low resolution resulting in very weak boundaries.

The limitations of the presented imaging modalities also make it difficult to use robust, accurate, fully automatic segmentation algorithms. This justifies to study semi-automatic algorithms, which include anatomical knowledge provided by the user as foreground and background scribbles or in terms of shape constraints.

1.5 Contribution

Our work is based on an interactive segmentation framework using Geodesic Active Contours (GAC) [54, 66, 70]. The aim of this thesis is to explore different types of shape constraints and extend the segmentation framework with these constraints. We study theoretical aspects as well as the impact of different shape constraints and combinations of constraints. We evaluate the constraints qualitatively and if possible quantitatively. We study global shape constraints based on image moments and a star convexity constraint as well as model specific shape constraints built from training images. These two types of shape constraints are shown to be useful for two major medical applications, sinus floor augmentation segmentation and vertebrae segmentation.

Sinus Floor Augmentation Segmentation[1]

Maxillary sinus floor augmentation is an operation to increase the amount of bone tissue in the upper jaw bone by putting grafting material in the maxillary sinus. This is often necessary to be able to set dental implants. Experts are interested in segmenting and quantifying grafting material in CT images and monitoring the decrease of grafting material over time. Until now, the segmentation is performed manually by drawing the boundaries of the grafting material in every axial slice [43]. However, it is not possible to assess the whole object in 3D and thus the manual method is prone to errors. In general, sinus floor augmentation segmentation is a very difficult task. On the one hand, different grafting materials are composed in different ways and thus appear with different HU as illustrated in Fig. 1.2. This does not allow to use a general intensity model for the prediction of grafting material. On the other hand, we often face weak or no boundaries due to presence of mucous tissue in the maxillary sinus as well as the similarity and connection to other bone structures.

Although we have to deal with lots of difficulties, it is possible to assume that the grafting material is a (star) convex shape. This motivates the use of global shape constraints. In this thesis, we show how we can use foreground and background scribbles to obtain a segmentation faster. Furthermore, we investigate if additional global shape constraints lower the amount of user interaction and further improve image segmentation.

---

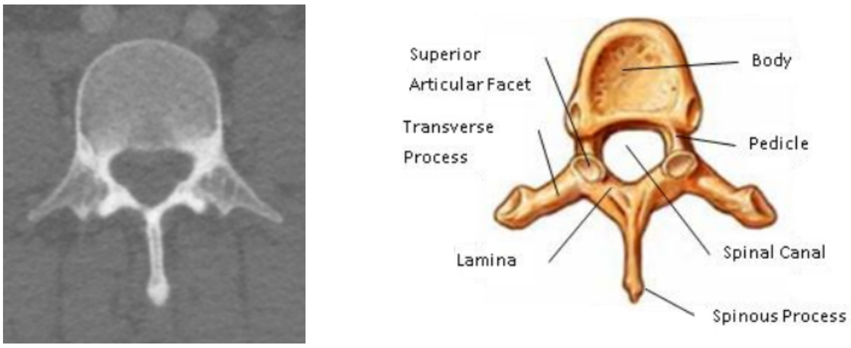
[1] Parts of this work were funded by a project of the Division of Prosthodontics, Restorative Dentistry, Periodontology and Implantology at the Department of Dentistry and Maxillofacial Surgery at the Medical University of Graz. We thank Dr. med. dent. Susanne Vogl for her support in this project.

Figure 1.2: Examples for different grafting materials. The first row shows axial cross-sections, the second row sagittal cross-sections of different grafting material appearances. Grafting material is indicated with yellow arrows. The left column shows a grafting material whose boundaries are difficult to estimate. The grafting material illustrated in the second column has many well-defined boundaries. However, the intensities are the same as surrounding bone tissue which makes it difficult to find the right boundary that encloses the grafting material.

Vertebrae Segmentation

As most spinal pathologies are related to vertebrae conditions, the development of methods for accurate and objective vertebrae segmentation in medical images represents an important and challenging research area. Although bone tissue has characteristic intensity values and good contrast to other tissues in CT images (see Fig. 1.3a), low-level image segmentation methods are not suitable for spine segmentation. Challenges for vertebrae segmentation are small intervertebral discs with a weak boundary to enclosing vertebrae, degenerative bone disease regions, osteophytes and connected ribs. Manual segmentation of vertebrae is tedious and too time consuming to be used in clinical practice. Semi-automatic methods often still require much user interaction because foreground and background scribbles have to be drawn for each individual vertebrae. This substantiates the use of shape prior information.

Due to the complex appearance of vertebral substructures as illustrated in Fig. 1.3b, global shape constraints are not sufficient and a more specific shape model is needed



(a) Axial CT slice of a vertebra **(b)** Anatomy of a vertebra

Figure 1.3: Illustrations of a vertebra in an axial CT image in (a) and the complex anatomy of a vertebra in (b). Image (b) is taken from [72].

which requires training data. There exists a database containing ten CT images with corresponding reference segmentations for each individual vertebra [73]. In this thesis, we show how to build a specific shape model based on the available data set and incorporate it into the segmentation framework. We set up an algorithm that minimizes the amount of user interaction to single point initialization in the vertebral bodies. This initialization step can be replaced by fully automatic localization methods that do not require any user interaction. However, this is beyond the scope of this thesis. We refer to e.g. [32] for more details.

1.6 Outline

This thesis is organized as follows: In Chapter 2 we review common variational methods in imaging. We introduce image denoising as well as different types of regularizations. Furthermore, we review variational methods for image segmentation starting from snakes to recent interactive, convex variational methods based on GAC that are used to build the basic framework for image segmentation used throughout this thesis. At the end of that chapter, we show how continuous variational problems can be solved numerically. Important mathematical tools to solve convex variational problems are introduced in Chapter 3. The basic segmentation framework is extended with shape constraints that are handled theoretically in Chapter 4. The impact of various shape constraints and the application to medical segmentation tasks such as sinus floor augmentation segmentation and vertebrae segmentation are evaluated in Chapter 5. Chapter 6 concludes the presented shape constraints and gives an outlook to future work and possible enhancements of the algorithms.

2

Variational Methods in Imaging

Contents

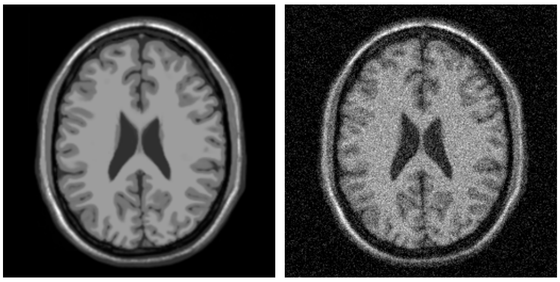
During the last years, variational methods have become state of the art methods to solve inverse problems in computer vision. Examples for inverse problems are image segmentation, denoising, motion correction, Computed Tomography (CT) reconstruction, optical flow and many others. However, these problems are often *ill-posed.* The opposite of ill-posed problems are well-posed problem defined by Hadamard [36]. The definition of well-posed problems states that

- a solution exists
- the solution is unique
- and the solution is stable, i.e. small changes in the data result in small changes in the solution

Inverse problems are not trivial to solve. Thus, we have to restrict the space of possible solutions, i.e., make assumptions on the solution itself by incorporating some regularization or prior knowledge. Tikhonov [64] proposed to write this as an (variational) optimization problem:

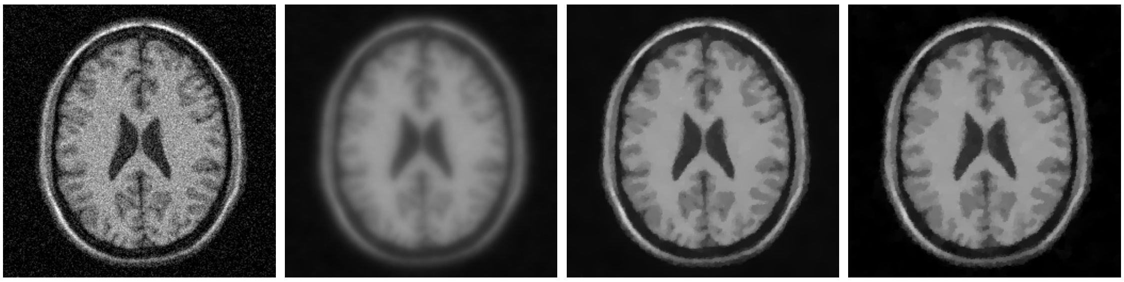
$$\min_u \mathcal{R}(u) + \lambda\, \mathcal{D}(u, f). \tag{2.1}$$

The data term $\mathcal{D}(u, f)$ models the relationship between a given observation f and the solution u, while the regularization term $\mathcal{R}(u)$ models prior assumptions made on the solution u. The parameter λ is adapted such that a good trade-off between regularization term and data term is found.



(a) Original image **(b)** Degraded image f

Figure 2.1: Original image u_{true} (a) degraded by 5% additive Gaussian noise (b).



(a) Noisy image **(b)** Tikhonov **(c)** ROF **(d)** TV-L^1

Figure 2.2: Comparison of different denoising algorithms. Tikhonov's method blurs the image. There is no remarkable difference between ROF and TV-L^1 model. Both methods remove noise while preserving image edges.

In this chapter, we show how various ill-posed image processing problems can be formulated as variational problems. We review common models used in image denoising. This is an important preprocessing step for image segmentation that we will also use in our framework. We introduce the Total Variation (TV) semi-norm as regularization. Furthermore, we give an introduction to interactive image segmentation and formulate a basic variational model that we will extend in Chapter 4.

2.1 Image Denoising

First, let us assume that an image u_{true} is degraded by additive, Gaussian noise n as depicted in Fig. 2.1. The noise model is then defined as:

$$f = u_{true} + n. \tag{2.2}$$

The inverse problem is stated as follows: Given the observed, degraded image f, how can we find the true solution u_{true}, when the noise n is unknown? We relate to three important denoising models: Tikhonov, TV-L^2 (ROF) and TV-L^1 denoising. A comparison of the three models is shown in Fig. 2.2.

2.1.1 Tikhonov Model

Tikhonov proposed to restore the image u by solving following variational problem [64, 65]:

$$\min_u \underbrace{\frac{1}{2}\int_\Omega |\nabla u|^2 \,\mathrm{d}\mathbf{x}}_{\mathcal{R}(u)} + \underbrace{\frac{\lambda}{2}\int_\Omega (u-f)^2 \,\mathrm{d}\mathbf{x}}_{\mathcal{D}(u,f)} \tag{2.3}$$

where Ω denotes the image domain. The functional is composed of a quadratic regularization term and a quadratic data fidelity term. The data fidelity term ensures that the solution u is close to the observed image f. An example denoising result is depicted in Fig. 2.2b. This denoising model is closely related to Gaussian deblurring which can be explained by Bayesian theory.

Bayesian Theory

Inverse problems and their applications to image processing have been studied from a probabilistic point of view [51]. The aim is to find the denoised image u that gives the maximum probability based on the observation f formulated as posterior probability $p(u|f)$:

$$u^* = \arg\max_u \{p(u|f)\}\,. \tag{2.4}$$

This is also known as Maximum A Posteriori (MAP) estimation. The posterior probability $p(u|f)$ can be expressed in terms of the a priori probability $p(u)$ and the conditional probability $p(f|u)$ by Bayes' theorem:

$$p(u|f) = \frac{p(u)p(f|u)}{p(f)} \tag{2.5}$$

where the factor $p(f)$ is a normalization factor. This normalization factor can be neglected in the maximization of the posterior probability $p(u|f)$. The conditional probability $p(f|u)$ states how well the observed function f can be described by the solution u and is often referred to as likelihood or data model. In the case of image denoising, we assume that the observed image f is degraded by additive Gaussian noise with zero mean and σ^2 variance. Hence, the conditional probability can be formulated in terms of independently and identically distributed Gaussian normal distributions:

$$p(f|u) = \prod_{u\in\Omega} \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(u-f)^2}{2\sigma^2}}\,. \tag{2.6}$$

To model the a priori probability, we use the Tikhonov regularization $|\nabla u|^2$ and assume again a Gaussian model with variance ν^2:

$$p(u) = \prod_{u \in \Omega} \frac{1}{\sqrt{2\pi}\,\nu} e^{-\frac{|\nabla u|^2}{2\nu^2}}. \tag{2.7}$$

Using Bayes' theorem and neglecting the scaling factor $p(f)$, since we will be interested in maximizing $p(u|f)$ later, we obtain:

$$p(u|f) = \prod_{u \in \Omega} \frac{1}{2\pi\nu\sigma} e^{-\frac{(u-f)^2}{2\sigma^2} - \frac{|\nabla u|^2}{2\nu^2}}. \tag{2.8}$$

This equation is related to the Boltzmann distribution (Gibbs distribution) known from thermodynamics:

$$p(u|f) \propto e^{-\frac{E(u)}{k_B T}} \tag{2.9}$$

where $E(u)$ is the free energy, k_B is the Boltzmann constant and T is the temperature. Maximizing the posterior probability is the same as minimizing the energy $E(u)$:

$$\max_u p(u|f) = \max_u e^{-\frac{E(u)}{k_B T}} = \min_u \frac{E(u)}{k_B T} = \min_u E(u). \tag{2.10}$$

In the case of denoising, the energy $E(u)$ is given by:

$$E(u) = \frac{1}{2\sigma^2} \int_\Omega (u-f)^2 \,\mathrm{d}\mathbf{x} + \frac{1}{2\nu^2} \int_\Omega |\nabla u|^2 \,\mathrm{d}\mathbf{x} = \frac{\lambda}{2} \int_\Omega (u-f)^2 \,\mathrm{d}\mathbf{x} + \frac{1}{2} \int_\Omega |\nabla u|^2 \,\mathrm{d}\mathbf{x} \tag{2.11}$$

where we set $\lambda = \frac{\nu^2}{\sigma^2}$. This is exactly Eqn. (2.3).

2.1.2 Total Variation Regularization with L^2 Data Term

Rudin, Osher and Fatemi (ROF) were one of the first who introduced TV in computer vision [58]. The difference to the Tikhonov model is that the regularization term is not quadratic. This term is known as TV semi-norm.

Total Variation Regularization

The TV semi-norm is defined as the L^1 norm of the L^2 norm of the image gradient $\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)$:

$$\mathrm{TV}(u) = \int_\Omega |\nabla u| \,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2} \,\mathrm{d}\mathbf{x}, \qquad u \in C^1. \tag{2.12}$$

Here, we assume that u is smooth, i.e., ∇u exists. The TV semi-norm can also be viewed geometrically. It minimizes the perimeter of all level sets s of u independently expressed by the co-area formula [30]:

$$\mathrm{TV}(u) = \int\limits_{-\infty}^{\infty} \mathrm{Per}\left(\{u > s\}\,;\Omega\right)\,\mathrm{d}s. \tag{2.13}$$

The denoising problem proposed by Rudin, Osher and Fatemi combines the TV regularization with an L^2 data term and is referred to as TV-L^2 or ROF model:

$$\min_u \int_\Omega |\nabla u|\,\mathrm{d}\mathbf{x} \qquad \text{s.t.} \qquad (u-f)^2 = \sigma^2. \tag{2.14}$$

They propose that the solution u and the observation f differ exactly by Gaussian noise of variance σ^2. However, this formulation is a non-convex constrained optimization problem. Chambolle et al. [16] proposed a convex formulation by replacing the equality constraint by an inequality constraint:

$$\min_u \int_\Omega |\nabla u|\,\mathrm{d}\mathbf{x} \qquad \text{s.t.} \qquad (u-f)^2 \leq \sigma^2. \tag{2.15}$$

They showed that this equation can be expressed by following strictly convex formulation

$$\min_u \underbrace{\int_\Omega |\nabla u|\,\mathrm{d}\mathbf{x}}_{\mathcal{R}(u)} + \frac{\lambda}{2} \underbrace{\int_\Omega (u-f)^2\,\mathrm{d}\mathbf{x}}_{\mathcal{D}(u,f)} \tag{2.16}$$

where λ regularizes the trade-off between data term and regularization term. This small change in the Tikhonov model leads to an important enhancement of image features. While edges are preserved, homogeneous regions are flattened. The amount of smoothing depends on the choice of the regularization parameter λ. Figure 2.2c illustrates an example for ROF denoising. Edges are clearly visible. We can also see many piecewise constant regions which is a drawback of the TV semi-norm, also known as staircaising artifact [50]. This effect could be overcome by using Total Generalized Variation [12].

2.1.3 Total Variation Regularization with L^1 Data Term

Until now, we have replaced the regularization term by an L^1 norm. It is also possible to replace the quadratic L^2 data fidelity term of Eqn. (2.16) by an L^1 norm yielding the TV-L^1 model [18, 50]:

$$\min_u \underbrace{\int_\Omega |\nabla u|\,\mathrm{d}\mathbf{x}}_{\mathcal{R}(u)} + \lambda \underbrace{\int_\Omega |u-f|\,\mathrm{d}\mathbf{x}}_{\mathcal{D}(u,f)}. \tag{2.17}$$

Compared to Eqn. (2.16), this model is not strictly convex, i.e., the globally optimal solution is not unique. More details on convex optimization can be found in Chapter 3. As illustrated in Fig. 2.2d, the difference to the ROF model seems negligible. Nevertheless, the TV-L^1 model has two nice properties. Firstly, it is contrast invariant. The second property is that it allows to remove salt-and-pepper noise [50].

2.2 From Snakes to Convex Variational Image Segmentation

Not only denoising, but also many image segmentation tasks are formulated as energy minimization problems. The most prominent example are active contours. The goal of active contours is to evolve a deformable curve towards object boundaries based on forces. A force is commonly modeled using image gradient information. In this section, we give an overview of important active contour models and introduce a non-smooth, convex variational image segmentation model using prior user knowledge. This model is used as basic framework for this thesis.

2.2.1 Active Contour Model / Snake Model

The idea of interactive variational image segmentation goes back to the famous active contour or snake model introduced by Kass [41] in 1988. Snakes are parametrized curves $C(x) = (x(s), y(s)) : [0,1] \mapsto \mathbb{R}^2$ which are evolved towards salient image features such as edges. The goal is to minimize the snakes energy defined by internal and external forces:

$$E(C(s)) = \int\limits_0^1 E_{internal}(C(s))\,\mathrm{d}s + \int\limits_0^1 E_{external}(C(s))\,\mathrm{d}s. \tag{2.18}$$

Internal forces $E_{internal}(C(s))$ describe the behavior of the curve $C(s)$:

$$E_{internal}(C(s)) = \frac{1}{2}\alpha|C'(s)|^2 + \frac{1}{2}\beta|C''(s)|^2. \tag{2.19}$$

The first derivative of $C(s)$ controlled by the parameter α defines the amount of elasticity, i.e., defines how much the curve can change in length. The curvature is characterized by the second derivative and β. Setting β to zero lets the curve develop corners. Hence, α and β control size and shape, respectively, of the snake. Furthermore, snakes are guided by external forces $E_{external}(C(s))$ such as image intensities, edges (gradient information) or zero-crossings as well as user constraints. Although snakes are widely used in medical image analysis, the traditional snakes model has a number of drawbacks. Firstly, the initialization is very critical and has to be done close to the optimal solution. Snakes can easily get stuck in local minima as the optimization problem is not convex. Furthermore, the zero set is a trivial solution. Secondly, it is not possible to handle topological changes, i.e., we cannot segment two objects having only one contour. Traditional snakes cannot capture concavities, which can be done by incorporating gradient vector fields as external force as proposed in [71]. Other problems are that traditional snakes need a lot of parameter tuning and the final solution strongly depends on the parametrization of the curve.

2.2.2 Geodesic Active Contours

Caselles et al. [14] proposed an improved snake model that is invariant to curve parametrization and incorporates geodesic information. This can be achieved by minimizing the Geodesic Active Contours (GAC) energy defined as

$$\min_C E_{GAC}(C) = \min_C \int\limits_0^{L(C)} g\left(\|\nabla I(C(s))\|\right) \, \mathrm{d}s \tag{2.20}$$

where $L(C)$ is the length of the curve and $g \in [0,1]$ is an edge function depending on the input image gradient ∇I. The objective is to weight each Euclidean length element ds by the edge function g to estimate a new, minimal curve length. Looking at this problem in 3D, we try to find the minimal surface. Common choices are $g(\mathbf{x}) = \frac{1}{1+\beta\|\nabla I(x)\|^2}$ or $g(\mathbf{x}) = e^{-\alpha\|\nabla I(x)\|^\beta}$, for parameters $\alpha, \beta > 0$. Nevertheless, this method is prone to get stuck in local minima due to the level set framework used for solving the GAC model [14]. Again, a trivial solution is the zero set.

2.2.3 Weighted Total Variation

Bresson et al. [13] reformulated the GAC energy in Eqn. (2.20) leading to the TV semi-norm weighted by the edge function g

$$TV_g(u) = \int_\Omega g(\mathbf{x})|\nabla u| \, \mathrm{d}\mathbf{x} = \int\limits_0^{L(C)} g(\mathbf{x}) \, \mathrm{d}s = E_{GAC}(C). \tag{2.21}$$

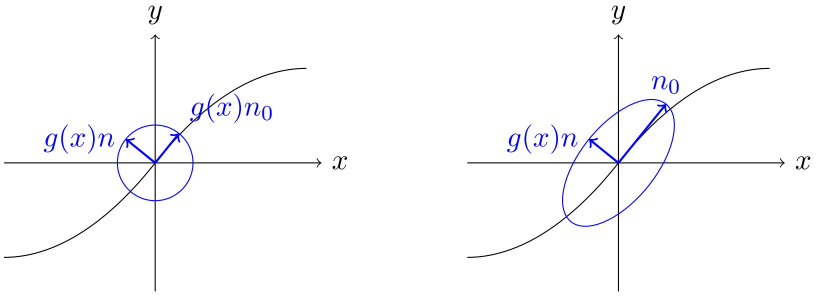
In this formulation, the weighted TV semi-norm incorporates only the image gradient magnitude. To be able to segment elongated and thin structures, Reinbacher et al. [54] replaced the edge function $g(\mathbf{x})$ by a diffusion tensor $D(\mathbf{x})$ as proposed by [69]:

$$TV_{g,aniso}(u) = \int_\Omega |D(\mathbf{x})^{\frac{1}{2}} \nabla u| \, \mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\nabla u^T D(\mathbf{x}) \nabla u} \, \mathrm{d}\mathbf{x} \tag{2.22}$$

with

$$D^{\frac{1}{2}}(\mathbf{x}) = g(\mathbf{x}) n n^T + n_0 n_0^T + n_1 n_1^T \tag{2.23}$$

where $n = \frac{\nabla I}{\|\nabla I\|}$ is the normalized image gradient, n_0 is the tangent to the edge and $n_1 = n \times n_0$. The diffusion tensor describes an ellipsoid whose major axes point along n_0 and n_1, while the minor axis scaled with $g(\mathbf{x})$ points along the edge direction n in three



(a) Isotropic scaling **(b)** Anisotropic scaling

Figure 2.3: Difference between isotropic and anisotropic scaling.

dimensions. Setting

$$D^{\frac{1}{2}}(\mathbf{x}) = \operatorname{diag}\left(g(\mathbf{x})\right) \tag{2.24}$$

yields the isotropic weighted TV semi-norm as defined in Eqn. (2.21):

$$\mathrm{TV}_g(u) = \int_\Omega \sqrt{g(\mathbf{x})^2 \nabla u^T \nabla u}\, \mathrm{d}\mathbf{x} = \int_\Omega g(\mathbf{x}) |\nabla u|\, \mathrm{d}\mathbf{x}. \tag{2.25}$$

For the isotropic case, we scale ∇u uniformly with $g(\mathbf{x})$. The different scalings for two dimensions are depicted in Fig. 2.3. Using anisotropic scaling favors ∇u to be directed along an edge.

To obtain a solution u based on the weighted TV semi-norm, Bresson et al. [13] proposed to minimize

$$\min_u TV_g(u) \qquad \text{s.t.} \qquad u \in \{0,1\} \tag{2.26}$$

where the segmentation u is a binary function $u \in \{0,1\}$, hence, no convex set. In order to obtain a globally optimal solution for the GAC model, u is varied smoothly between $[0,1]$. The final segmentation is then obtained by taking an arbitrary level set between $[0,1]$. As we now have a convex functional, the zero set is always a global minimizer. Thus, we need to incorporate additional constraints:

$$\min_u TV_g(u) + \text{constraints} \qquad \text{s.t.} \qquad u \in [0,1]. \tag{2.27}$$

Bresson et al. [13] proposed also a variational model for image segmentation combining the weighted TV semi-norm with an L^1 data fidelity term:

$$\min_u TV_g(u) + \lambda \int_\Omega |u - f|\, \mathrm{d}\mathbf{x} \qquad \text{s.t.} \qquad u \in [0,1] \tag{2.28}$$

where f is a given input image.

2.2.4 Convex Variational Image Segmentation based on User-Specified Prior Knowledge

Unger et al. [66] proposed to include prior knowledge in terms of manual user constraints in the variational framework by introducing a spatially varying regularization parameter $\lambda(\mathbf{x})$:

$$\min_u TV_g(u) + \int_\Omega \lambda(\mathbf{x})|u - f|\,\mathrm{d}\mathbf{x} \qquad \text{s.t.} \qquad u \in [0,1] \tag{2.29}$$

where $f \in [0,1]$ contains manual user constraints. Values $f = 1$ are foreground and $f = 0$ background seed regions, respectively. The spatially varying parameter $\lambda(\mathbf{x})$ defines the behavior of the algorithm. If $\lambda = 0$, the solution u is only influenced by the weighted TV semi-norm, hence, the surrounding image edges. This means that the pure GAC energy is minimized. If $\lambda = \infty$, the algorithm is forced to set $u = f$. Setting $0 < \lambda < \infty$ defines the trade-off between regularization and data fidelity term.

Throughout this thesis, we use and extend following segmentation method, which has been used by Reinbacher et al. [54] and Werlberger et al. [70] previously:

$$\min_u \underbrace{TV_g(u)}_{\substack{\text{anisotropic}\\ \text{or}\\ \text{isotropic}}} + \lambda \int_\Omega u \cdot w\,\mathrm{d}\mathbf{x} \qquad \text{s.t.} \qquad u \in [0,1]. \tag{2.30}$$

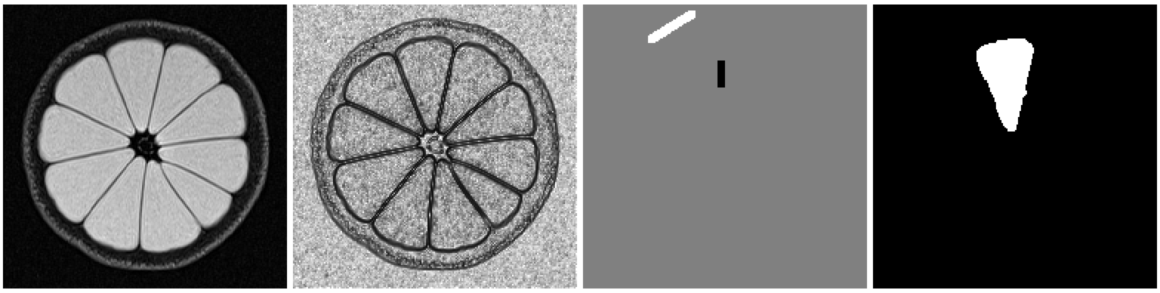
This can be seen as a continuous graph cut [9]. The edge function $g(\mathbf{x})$ is defined as

$$g(\mathbf{x}) = e^{-\alpha\|\nabla I(\mathbf{x})\|^\beta},\ \alpha, \beta > 0. \tag{2.31}$$

Depending on the underlying segmentation problem, we can choose between isotropic and anisotropic scaling. An example segmentation is illustrated in Fig. 2.4. The function w is a function which accounts for constraints. To obtain a segmentation, one has to provide a valid function w, otherwise the segmentation u will be the zero set. The constraints in w have to be chosen as follows:

- **Pure GAC Energy:** If $w = 0$, the solution u is only influenced by the weighted TV semi-norm, hence, the surrounding image edges.
- **Foreground:** If $w < 0$, the algorithm tends to set $u = 1$ in the minimization scheme.
- **Background:** If $w > 0$, the algorithm tends to set $u = 0$ in the minimization scheme.

We consider three possibilities for the constraint function w. User constraints, weighting maps or a combination of both.



(a) Input **(b)** Edges $g(\mathbf{x})$ **(c)** Constraints w **(d)** Segmentation

Figure 2.4: This example shows a g-weighted TV based segmentation of a lemon using solely user constraints. The input image is illustrated in (a), the edge image in (b). Image (c) shows the manual user constraints, where black regions correspond to foreground and white regions to background, respectively. The final segmentation is depicted in (d).

2.2.4.1 User Constraints

Choosing $w = \infty$ or $w = -\infty$ forces the segmentation u to be 0 or 1, respectively, neglecting the edges in those areas. Practically, this can be done by drawing foreground and background scribbles on the image. An example for using pure user constraints is illustrated in Fig. 2.4. Figure 2.4c shows the user constraints. White regions correspond to background scribbles and black regions correspond to foreground scribbles. The segmentation result is mainly based on the surrounding edges depicted in Fig. 2.4b, i.e., only the GAC energy is minimized in regions where no user constraints are set.

2.2.4.2 Weighting maps

Values for w below infinity and regularized by the parameter λ have a weaker impact, they lead the segmentation to the final segmentation u. The lower the parameter λ is chosen, the smoother is the final segmentation. For gray value images, we can calculate a weighting map fulfilling the requirements for background and foreground by using the log-likelihood ratio between the probability that a pixel x belongs to the foreground distribution p_{fg} and the probability that it belongs to the background distribution p_{bg}

$$w = \log \frac{p_{bg}}{p_{fg}}. \tag{2.32}$$

To realize this, we build normalized foreground and background histograms based on a mask or user-defined scribbles on the input image. Each pixel is then classified according to Eqn. (2.32) where the probabilities p_{bg}, p_{fg} can be looked up in the histograms. The same scheme can be applied to color images where we use normalized three dimensional histograms to account for each color channel [66].

A generic weighting map for color images solely based on foreground information can be obtained by calculating Bhattacharyya distances as proposed by Donoser et al. [29]

resulting in a map between $[0, 1]$. The maximum value 1 is likely to be foreground. Thus, we have to invert the map and center it at zero to obtain a valid weighting map for our segmentation algorithm.

2.3 Numerical Solution of Continuous Variational Problems

Up to now, we have set up different variational problems for denoising and segmentation by means of continuous energy-minimization functionals $E(u)$:

$$\begin{aligned} E(u) &= \int_\Omega \mathcal{L}(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}))\,\mathrm{d}\mathbf{x} \\ &= \iiint\limits_\Omega \mathcal{L}(x, y, z, u(\mathbf{x}), u_x(\mathbf{x}), u_y(\mathbf{x}), u_z(\mathbf{x}))\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \end{aligned} \tag{2.33}$$

where $\mathcal{L}(\cdot)$ is the Lagrangian function. In order to obtain a numerical solution, the continuous formulations of the variational models have to be discretized [15]. On the one hand, we have to discretize the image domain Ω. On the other hand, we have to discretize the energy functional and thus the derivative operators that appear in the functional. The discretized energy functional is written as follows:

$$E(u) = \sum_i \sum_j \sum_k \mathcal{L}(\mathbf{i}, u_\mathbf{i}, (\nabla u)_\mathbf{i}) h_x h_y h_z \tag{2.34}$$

where $\mathbf{i} = (i, j, k)$. We assume that we operate on a 3D image with discrete image domain $\Omega_d : M \times N \times D$ defined on a Cartesian grid. The physical distance between pixels, also known as image spacing, is denoted as h_x, h_y, h_z. Especially in medical imaging, the pixel spacings h_x, h_y, h_z are of great importance because we often deal with anisotropic pixels. The discrete pixel locations are given by:

$$\begin{aligned} x_i &= i \cdot h_x, \quad 1 \le i \le M \\ y_j &= j \cdot h_y, \quad 1 \le j \le N \\ z_k &= k \cdot h_z, \quad 1 \le k \le D. \end{aligned} \tag{2.35}$$

We have to discretize two derivative operators that appear in the energy functional: The gradient of a scalar field u and the divergence of a vector field p. We denote the current position as $\mathbf{i} = (i, j, k)$. To realize the discrete gradient operator $\nabla u = (u_x, u_y, u_z)^T$ we use forward differences δ^+

$$(\nabla u)_\mathbf{i} = \begin{pmatrix} \delta_x^+ u_\mathbf{i} \\ \delta_y^+ u_\mathbf{i} \\ \delta_z^+ u_\mathbf{i} \end{pmatrix} \tag{2.36}$$

with Neumann boundary conditions

$$\begin{aligned}
\delta_x^+ u_{\mathbf{i}} &= \begin{cases} \frac{u_{i+1,j,k}-u_{i,j,k}}{h_x} & \text{if} \quad 1 \leq i < M \\ 0 & \text{if} \quad i = M \end{cases} \\
\delta_y^+ u_{\mathbf{i}} &= \begin{cases} \frac{u_{i,j+1,k}-u_{i,j,k}}{h_y} & \text{if} \quad 1 \leq j < N \\ 0 & \text{if} \quad j = N \end{cases} \\
\delta_z^+ u_{\mathbf{i}} &= \begin{cases} \frac{u_{i,j,k+1}-u_{i,j,k}}{h_z} & \text{if} \quad 1 \leq k < D \\ 0 & \text{if} \quad k = D \end{cases} .
\end{aligned} \tag{2.37}$$

The TV semi-norm is the discrete l_1 norm of the l_2 norm of the vector field ∇u. Hence, it is defined by replacing the integral of the continuous definition by a sum leading to:

$$\begin{aligned}
\|\nabla u\|_{TV} = \|\nabla u\|_{2,1} = \sum_{\mathbf{i}} |(\nabla u)_{\mathbf{i}}| = \\
= \sum_{\mathbf{i}} \sqrt{(\delta_x^+ u_{\mathbf{i}})^2 + (\delta_y^+ u_{\mathbf{i}})^2 + (\delta_z^+ u_{\mathbf{i}})^2} .
\end{aligned} \tag{2.38}$$

At this point, we also define the discrete g-weighted TV semi-norm that we use for our isotropic image segmentation model:

$$\begin{aligned}
\|\nabla_g u\|_{TV} = \|\nabla_g u\|_{2,1} = \sum_{\mathbf{i}} g_{\mathbf{i}} |(\nabla u)_{\mathbf{i}}| = \\
= \sum_{\mathbf{i}} g_{\mathbf{i}} \sqrt{(\delta_x^+ u_{\mathbf{i}})^2 + (\delta_y^+ u_{\mathbf{i}})^2 + (\delta_z^+ u_{\mathbf{i}})^2} .
\end{aligned} \tag{2.39}$$

The second derivative operator that we need to discretize is the divergence operator of a vector field $p = (p^x, p^y, p^z)$. It has to fulfill following condition:

$$\left\langle u, \nabla^T p \right\rangle = \langle u, -\operatorname{div} p \rangle . \tag{2.40}$$

The discrete divergence operator is realized by backward differences δ^- with Dirichlet boundary conditions:

$$\begin{aligned}
\operatorname{div} p_{\mathbf{i}} &= \delta_x^- p_{\mathbf{i}} + \delta_y^- p_{\mathbf{i}} + \delta_z^- p_{\mathbf{i}} \\
&= \begin{cases} \frac{p^x_{\mathbf{i}} - p^x_{i-1,j,k}}{h_x} & \text{if } 1<i<M \\ p^x_{\mathbf{i}} & \text{if } i=1 \\ -p^x_{i-1,j,k} & \text{if } i=M \end{cases} + \begin{cases} \frac{p^y_{\mathbf{i}} - p^y_{i,j-1,k}}{h_y} & \text{if } 1<j<N \\ p^y_{\mathbf{i}} & \text{if } j=1 \\ -p^y_{i,j-1,k} & \text{if } j=N \end{cases} + \begin{cases} \frac{p^z_{\mathbf{i}} - p^z_{i,j,k-1}}{h_z} & \text{if } 1<k<D \\ p^z_{\mathbf{i}} & \text{if } k=1 \\ -p^z_{i,j,k-1} & \text{if } k=D \end{cases} .
\end{aligned} \tag{2.41}$$

3

Convex Optimization

Contents

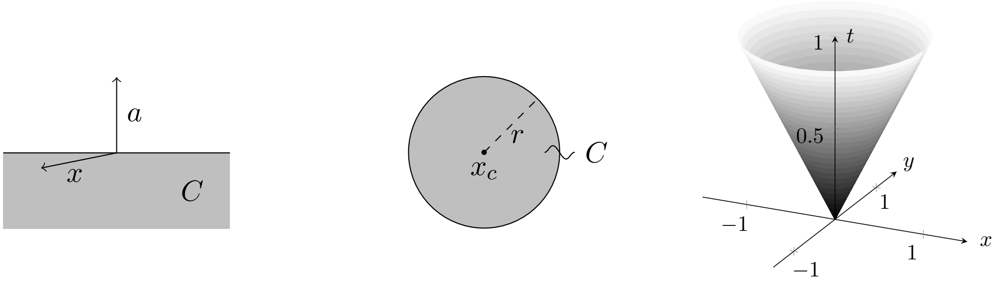
A basic convex optimization problem is defined as follows:

$$\begin{aligned} &\min f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, ..., p \\ & h_j(x) = 0, \quad j = 1, ..., q \end{aligned} \tag{3.1}$$

where $x \in \mathbb{R}^n$ is the optimization variable, $f_0(x) : \mathbb{R}^n \mapsto \mathbb{R}$ the objective function subject to constraints $f_i(x) \leq 0$ and $h_j(x) = 0$. Inequality constraint functions are denoted as $f_i(x) : \mathbb{R}^n \mapsto \mathbb{R}$, equality constraint functions as $h_j(x) : \mathbb{R}^n \mapsto \mathbb{R}$, respectively. If there are no constraints, i.e., $p = q = 0$, we have an unconstrained optimization problem. The optimal solution x^* of this problem is as follows:

$$x^* = \inf\{f_0(x) \,|\, f_i(x) \leq 0 \wedge h_j(x) = 0, \, \forall i, j\}. \tag{3.2}$$

In the following, we introduce basic tools of convex optimization that are necessary to solve various tasks in computer vision. Furthermore, we show algorithms to solve unconstrained and constrained optimization problems. More details on convex optimization can be found in [2, 4, 6, 49, 55].

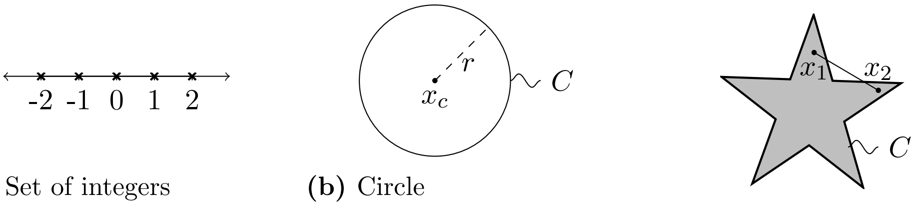


(a) Half-space
$C: \left\{x \,|\, a^T x \leq b\right\},\ a \neq 0$

(b) Euclidean ball
$C: \{x \,|\, \|x - x_c\| \leq r\}$

(c) Second order cone
$C: \{x \,|\, \|x\| \leq t\}$

Figure 3.1: Examples for convex sets



(a) Set of integers
$C: \{x \,|\, x \in \mathbb{Z}\}$

(b) Circle
$C: \{x \,|\, \|x - x_c\| = r\}$

(c) Star shape

Figure 3.2: Examples for non-convex sets

3.1 Convex Sets

Consider two points $x_1, x_2 \in C$ that lie in the set C. For convex sets C, it holds that all line segments between the two points again lie in C:

$$x_1, x_2 \in C, \quad \alpha x_1 + (1-\alpha)x_2 \in C, \quad \alpha \in [0,1]. \tag{3.3}$$

Examples for convex sets are depicted in Fig. 3.1. Counter examples are illustrated in Fig. 3.2.

3.2 Convex Functions

A function $f: C \mapsto \mathbb{R}$ is *convex* if the following inequality holds:

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2), \quad x_1, x_2 \in C,\, \alpha \in [0,1] \tag{3.4}$$

where C is a convex set. Strict convexity is given if and only if the above inequality is strict

$$f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha) f(x_2), \quad x_1, x_2 \in C,\, \alpha \in (0,1). \tag{3.5}$$

For differentiable functions, we have alternative characterizations of convexity.

First-order condition If f is differentiable over $\mathbb{R}^n$, the function f is convex if

$$f(y) \geq f(x) + (y-x)^T \nabla f(x), \quad \forall x, y, \in C. \tag{3.6}$$

Strict convexity is given if and only if

$$f(y) > f(x) + (y-x)^T \nabla f(x), \quad \forall x, y, \in C. \tag{3.7}$$

Hence, a convex function $f(y)$ can be underestimated by means of the first-order approximation $f(x) + (y-x)^T \nabla f(x)$.

Second-order condition For twice differentiable functions f over the subset $\mathbb{R}^n$, the following conditions only can prove convexity:

- If the Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in C$, then f is convex over the convex set C.
- If the Hessian matrix $\nabla^2 f(x)$ is positive definite for all $x \in C$, then f is strictly convex over the convex set C.
- If $f(x) = x^T A x$, f is convex over the convex set C, if and only if A is positive semidefinite and symmetric.

For convex functions, it holds that every local minimum is also a global minimum. If the strict inequality in Eqn. (3.5) holds, there exists only one global minimum. Hence, we conclude that the global optimum does not necessarily need to be unique. In the following, we list examples for convex functions:

- affine $a^T x + b \quad a \in \mathbb{R}^n, b \in \mathbb{R}$
- exponential $e^{ax}, \quad a \in \mathbb{R}$
- powers x^a on $\mathbb{R}_{++}, \quad a \geq 1 \vee a \leq 0$
- l_p norms $\|x\|_p = \left(\sum\limits_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, \quad p \geq 1$

Indicator Function To realize restrictions on optimization variables, the indicator function is used:

$$I_C(x) = \begin{cases} 0 & \text{if} \quad x \in C \\ \infty & \text{if} \quad x \notin C \end{cases}. \tag{3.8}$$

If the set C is a convex set, the indicator function itself is convex, too. Thus, we can write the optimization problem $f(x)$ restricting the variable x to the convex set C in terms of a constrained or an unconstrained optimization problem:

$$\min_{x \in C} f(x) = \min_{x} f(x) + I_C(x). \tag{3.9}$$

3.3 Smooth Convex Optimization

Let us now consider an unconstrained optimization problem of the form $\min\limits_{x} f(x)$ where the function $f(x)$ is smooth (continuously differentiable) and convex. We denote this class of functions $\mathcal{F}_L^{k,l}$. Functions of class $\mathcal{F}_L^{k,l}$ are $k-$times continuously differentiable. Furthermore, the $l-$th derivative is said to be Lipschitz continuous with Lipschitz constant L:

$$\|f^{(l)}(x_1) - f^{(l)}(x_2)\| \leq L\|x_1 - x_2\|. \tag{3.10}$$

In this thesis, we deal with functions of class $\mathcal{F}_L^{1,1}$. These functions are 1-time continuously differentiable and following inequality holds:

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L\|x_1 - x_2\|. \tag{3.11}$$

Gradient Descent

For unconstrained optimization problems $\min\limits_{x} f(x)$ and functions $f(x)$ of class $\mathcal{F}_L^{1,1}$, the solution can be obtained by the simple gradient descent method shown in Algorithm 1. The step size τ has to be chosen such that the algorithm converges. Nesterov [49] showed that for this class of problems $\mathcal{F}_L^{1,1}$ the algorithm converges with rate $\mathcal{O}\left(\frac{1}{k}\right)$ for a step size $\tau \in \left(0, \frac{2}{L}\right)$.

Algorithm 1 Gradient descent

```
Choose $x^0 \in \mathbb{R}^n$, $0 < \tau < \frac{2}{L}$.
for $k \geq 0$ do
    $x^{k+1} = x^k - \tau \nabla f(x^k)$
end for
```

3.4 Non-Smooth Convex Optimization

This section introduces a framework to handle possible non-smooth functions. Firstly, the concept of duality is introduced to handle functions with discontinuities. Furthermore, the proximal mapping and the proximal gradient descent method, which is a generalization of the basic gradient descent algorithm, are presented to solve optimization problems composed of smooth and non-smooth functions.

3.4.1 Legendre-Fenchel Transformation - An Introduction to Duality

Transformations are powerful tools to analyze functions in a different space. Similar to Fourier transform being important to analyze functions in the spectral domain, the Legendre-Fenchel transform is essential in convex analysis.

Let us first introduce the Legendre-Fenchel transform:

$$f^*(y) = \sup_{x \in \operatorname{dom} f} \langle x, y \rangle - f(x) \tag{3.12}$$

where $f(x)$ is an arbitrary (non-)convex function and $f^*(y)$ is the convex conjugate of $f(x)$. In the following we show why we need this abstract formulation and why it helps to shift the problem to a new problem in terms of another variable.

Duality Sometimes a problem is easier to solve if we look at it from a different perspective. Using the Legendre-Fenchel transform in Eqn. (3.12), we transform the primal problem $f(x)$ into the dual space. Analyzing this transform in more detail, we see that the convex conjugate $f^*(y)$ is defined as the supremum over linear functions, hence, it is always convex independent of the initial function $f(x)$. We map points x of $f(x)$ into slopes y of $f^*(y)$. Geometrically seen, the function $f^*(y)$ is the intersection with the y-axis for an optimal point $x_{opt} = 0$. To make this concept clear, let us have a closer look at an example. Let the primal function be the absolute function $f(x) = |x|$ which has a discontinuity at $x = 0$. Using Eqn. (3.12), we get the dual formulation:

$$f^*(y) = \sup_{x \in \operatorname{dom} f} \langle x, y \rangle - |x|. \tag{3.13}$$

Differentiating $f^*(y)$ with respect to x yields:

$$y = \frac{x}{|x|} = \operatorname{sgn}(x), \quad x \neq 0. \tag{3.14}$$

The gradient is not defined at $x = 0$. At this point, we have to introduce a generalization of the gradient for non-differentiable functions called the subgradient. In general, the gradient is the slope of a function, i.e., tangent to a function. At the discontinuity $x = 0$ we have several possibilities to define tangents. This set of tangents is called subdifferential $\partial f(x)$ at this point. A subgradient is an element of this set. For our example, the subdifferential is defined as follows:

$$\partial f(x) = \begin{cases} \operatorname{sgn}(x) & \text{if} \quad x \neq 0 \\ \in [-1, 1] & \text{if} \quad x = 0 \end{cases}. \tag{3.15}$$

The dual variable y can be any of these slopes:

$$y \in \partial f(x). \tag{3.16}$$

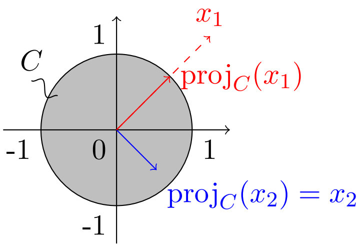


Figure 3.3: Euclidean projection of x_1 and x_2 onto the convex set representing the unit ball $\{C \,|\, \|x_i\| \leq 1,\, \forall i\}$.

If we consider a line with a slope $\notin [-1, 1]$, it is only tangent to the function at $y = \infty$. Thus, the convex conjugate equals the indicator function introduced in Eqn. (3.8):

$$f^*(y) = \begin{cases} 0 & \text{if} \quad |y| \leq 1 \\ \infty & \text{else} \end{cases} . \tag{3.17}$$

So far, we have defined the convex conjugate $f^*(y)$ of a function $f(x)$. Applying the Legendre-Fenchel transform 3.12 to $f^*(y)$ yields the bi-conjugate $f^{**}(x)$:

$$f^{**}(x) = \sup_{y \in \mathrm{dom}\, f} \langle y, x \rangle - f^*(y). \tag{3.18}$$

Furthermore, the Fenchel inequality always holds:

$$f^{**}(x) \leq f(x). \tag{3.19}$$

If $f(x)$ is already convex, $f^{**}(x)$ equals the primal problem $f(x)$. For non-convex functions $f(x)$, the bi-conjugate $f^{**}(x)$ is the largest convex envelope. Hence, we have a method to convexify non-convex functions.

3.4.2 Projection onto Convex Sets

Before we show how we can project onto an arbitrary number of convex sets, let us first consider the Euclidean projection of a point $\hat{x}$ onto a single convex set C. The projection operator is defined as follows:

$$\mathrm{proj}_C(\hat{x}) = \arg\min_{x \in C} \frac{1}{2}\|x - \hat{x}\|^2 = \arg\min_{x} \frac{1}{2}\|x - \hat{x}\|^2 + I_C(x) \tag{3.20}$$

where $I_C(x)$ denotes the indicator function of the convex set C defined in Eqn. (3.8). Note that the solution of the projection operator is unique. An example for a Euclidean projection onto a unit ball $\left\{C : \|x\| \leq 1,\, x \in \mathbb{R}^N\right\}$ is illustrated in Fig. 3.3 for $N = 2$. If we want now to project onto an intersection of n convex sets, Eqn. (3.20) turns into:

$$\operatorname{proj}_{C=\cap_{i=1}^{n} C_i}(\hat{x}) = \arg\min_{x} \frac{1}{2}\|x-\hat{x}\|^2 + \sum_{i=1}^{n} I_{C_i}(x). \tag{3.21}$$

Boyle and Dykstra proposed a method to solve the stated problem in [11]. The solution of Eqn. (3.21) is obtained by generating two sequences $\{\hat{x}_i^k\}$ and $\{h_i^k\}$. The current iterates $\{\hat{x}_i^k\}$ are stored in a vector X according to following scheme:

$$X = \begin{bmatrix} \hat{x}_1^{k+1} & \hat{x}_2^{k+1} & \dots & \hat{x}_i^k & \dots & \hat{x}_{n-1}^k & \hat{x}_n^k \end{bmatrix}. \tag{3.22}$$

The iterative projection algorithm is depicted in Algorithm 2.

Algorithm 2 Dykstra's projection algorithm (primal)

```
Choose $\hat{x}_n^0 = \hat{x}$. Set $h_i^0 = 0$, $i = 1, ..., n$.
for $k \to \infty$ do
  $\hat{x}_1^{k+1} = \hat{x}_n^k$
  for $i = 1, ..., n$ do
    $\hat{x}_i^{k+1} = \operatorname{proj}_{C_i}(\hat{x}_{i-1}^{k+1} - h_i^k)$
    $h_i^{k+1} = \hat{x}_i^{k+1} - (\hat{x}_i^k - h_i^k)$
  end for
end for
Solution $x = \hat{x}_n^k$, $k \to \infty$
```

The problem stated in Eqn. (3.21) is usually solved via duality [31] because Dykstra's algorithm can be explained in a more convenient way in terms of the coordinate descent [4]. Gaffke et al. [31] showed that using duality, the convergence can be proved easier compared to [11]. For the dual formulation of Dykstra's algorithm, we have to introduce dual variables y_i, $i = 1, ..., n$ accounting for each individual convex set I_{C_i}. Equation (3.21) turns into:

$$\min_{x} \max_{\{y_i\}} \frac{1}{2}\|x-\hat{x}\|^2 + \langle x, \sum_{i=1}^{n} y_i \rangle - \sum_{i=1}^{n} I_{C_i}^*(y_i) \tag{3.23}$$

where $I_{C_i}^*(y)$ denotes the convex conjugate of the indicator function $I_{C_i}(x)$. Minimizing with respect to x leads to:

$$x - \hat{x} + \sum_{i=1}^{n} y_i = 0 \Leftrightarrow x = \hat{x} - \sum_{i=1}^{n} y_i. \tag{3.24}$$

Our optimization problem now turns into:

$$\max_{\{y_i\}} \frac{1}{2}\|\hat{x} - \sum_{i=1}^{n} y_i - \hat{x}\|^2 + \langle \hat{x}, \sum_{i=1}^{n} y_i \rangle - \|\sum_{i=1}^{n} y_i\|^2 - \sum_{i=1}^{n} I^*_{C_i}(y_i) =$$

$$\max_{\{y_i\}} -\frac{1}{2}\|\sum_{i=1}^{n} y_i\|^2 + \langle \hat{x}, \sum_{i=1}^{n} y_i \rangle - \sum_{i=1}^{n} I^*_{C_i}(y_i).$$

Let us define the matrix $Y = \begin{bmatrix} y_1 & y_2 & ... & y_{n-1} & y_n \end{bmatrix}$ holding all y_i. We introduce a cyclic iterative scheme for alternating optimization of the dual variables y_i while keeping the other $y_j, j \neq i$ fixed. Therefore, we have to extend the matrix Y with the current iterates k:

$$Y = \begin{bmatrix} y_1^{k+1} & y_2^{k+1} & ... & y_i^k & ... & y_{n-1}^k & y_n^k \end{bmatrix}. \tag{3.25}$$

Now, we are able to solve the following maximization problem. Note that constant terms can be neglected in the optimization process.

$$\max_{y_i^k} -\frac{1}{2}\|y_i^k + \sum_{\substack{j=1\\j\neq i}}^{n} y_j\|^2 + \langle \hat{x}, y_i^k + \underbrace{\sum_{\substack{j=1\\j\neq i}}^{n} y_j}_{const.} \rangle - I^*_{C_i}(y_i^k) - \underbrace{\sum_{\substack{j=1\\j\neq i}}^{n} I^*_{C_j}(y_j)}_{const.} =$$

$$\min_{y_i^k} \frac{1}{2}\|y_i^k + \sum_{\substack{j=1\\j\neq i}}^{n} y_j\|^2 - \langle \hat{x}, y_i^k \rangle + I^*_{C_i}(y_i^k) =$$

$$\min_{y_i^k} \frac{1}{2}\|y_i^k\|^2 + \langle y_i^k, \sum_{\substack{j=1\\j\neq i}}^{n} y_j \rangle + \underbrace{\frac{1}{2}\|\sum_{\substack{j=1\\j\neq i}}^{n} y_j\|^2}_{const.} - \langle \hat{x}, y_i^k \rangle + I^*_{C_i}(y_i^k) =$$

$$\min_{y_i^k} \frac{1}{2}\|y_i^k\|^2 - \langle y_i^k, \hat{x} - \sum_{\substack{j=1\\j\neq i}}^{n} y_j \rangle + \frac{1}{2}\|\hat{x} - \sum_{\substack{j=1\\j\neq i}}^{n} y_j\|^2 \underbrace{- \frac{1}{2}\|\hat{x} - \sum_{\substack{j=1\\j\neq i}}^{n} y_j\|^2}_{const.} + I^*_{C_i}(y_i^k) =$$

$$\min_{y_i^k} \frac{1}{2}\|y_i^k - \left(\hat{x} - \sum_{\substack{j=1\\j\neq i}}^{n} y_j \right)\|^2 + I^*_{C_i}(y_i^k) =$$

$$\min_{y_i^k} \frac{1}{2}\|y_i^k - z\|^2 + I^*_{C_i}(y_i^k), \qquad z = \hat{x} - \sum_{\substack{j=1\\j\neq i}}^{n} y_j. \tag{3.26}$$

Equation (3.26) is exactly the conjugated projection operator which is the solution for the update y_i^{k+1}:

$$y_i^{k+1} = \mathrm{proj}_{C_i^*}(z) = \arg\min_{y_i^k} \frac{1}{2}\|y_i^k - z\|^2 + I_{C_i}^*(y_i^k). \tag{3.27}$$

Moreau's identity, which is introduced in Eqn. (3.31) in a more general way,

$$z = \mathrm{proj}_{C_i^*}(z) + \mathrm{proj}_{C_i}(z) \tag{3.28}$$

expresses the dual projection operator in terms of the primal projection operator. Hence, we can calculate the solution without calculating the conjugated projection and we obtain the final solution which equals exactly Eqn. 9 of Gaffke [31]:

$$y_i^{k+1} = z - \mathrm{proj}_{C_i}(z). \tag{3.29}$$

Finally, we obtain Algorithm 3 for the projection of a point $\hat{x}$ onto an intersection of convex sets $C = \cap_{i=1}^n C_i$ based on alternating minimization using duality. In practice, this projection has to be performed at every iteration of an outer optimization algorithm (e.g. primal-dual Algorithm 5 explained later). Assuming that x does not change much during one iteration of the outer algorithm, the last set of dual variables y should be used as initialization in the next projection steps for faster convergence. This initialization set of variables is also called hot starting points.

Algorithm 3 Dykstra's projection algorithm (dual)

```
Introduce dual variables $y_i^0$, $i = 1, ..., n$.
for $k \to \infty$ do
  for $i = 1, ..., n$ do
    $y_i^{k+1} = \hat{x} - \sum_{j=1, j\neq i}^{n} y_j - \mathrm{proj}_{C_i}(\hat{x} - \sum_{j=1, j\neq i}^{n} y_j)$
  end for
end for
Solution $x = \hat{x} - \sum_{i=1}^{n} y_i^k, k \to \infty$
```

3.4.3 Proximal Operator

The projection operator leads to the idea of generalizing it to the proximal operator. This is done by replacing the indicator function I_C by an arbitrary convex function h:

$$\mathrm{prox}_{\tau \cdot h}(\hat{x}) = \arg\min_{x} \frac{1}{2}\|x - \hat{x}\|^2 + \tau \cdot h(x). \tag{3.30}$$

As the projection operator, the proximal operator also has a unique solution. Examples for the proximal operator are:

- $h(x) = 0 \Rightarrow \operatorname{prox}_{\tau \cdot h}(\hat{x}) = \hat{x} = x$
- Pointwise Euclidean projection onto the unit ball for $\hat{x} \in \mathbb{R}^{N \times D}$
  $h(x) = I_{\|\hat{x}\|_{2,\infty} \leq 1}(\hat{x}) \Rightarrow \operatorname{prox}_{\tau \cdot h}(\hat{x})_i = \frac{\hat{x}_i}{\max(1, \|\hat{x}_i\|)}$
- Shrinkage (soft-threshold) operator
  $h(x) = \|x\|_{l_1} \Rightarrow \operatorname{prox}_{\tau \cdot h}(\hat{x})_i = \max(0, |\hat{x}_i| - \tau)\operatorname{sgn}(\hat{x}_i)$

The Moreau identity is a nice property which relates the proximal mappings with respect to a function $h(x)$ and its conjugate $h^*(x)$:

$$x = \operatorname{prox}_{\tau \cdot h}(x) + \tau \operatorname{prox}_{\frac{h^*}{\tau}}\left(\frac{x}{\tau}\right). \tag{3.31}$$

This is useful because it is often easier to express the proximal mapping of a function $h(x)$ in terms of the proximal mapping of its conjugate function $h^*(x)$ and vice versa instead of calculating the proximal mapping directly.

3.4.4 Proximal Gradient Descent

The basic gradient descent presented in Algorithm 1 can be generalized to the proximal gradient descent. We consider the following composite, unconstrained optimization problem:

$$\min_x f(x) + h(x) \tag{3.32}$$

where $f(x)$ is convex with Lipschitz continuous gradient and $h(x)$ is a convex function whose proximal mapping is inexpensive to compute.

Algorithm 4 Proximal gradient descent

```
Choose $x^0 \in \mathbb{R}^n$, $0 < \tau \leq \frac{1}{L}$.
for $k \geq 0$ do
    $x^{k+1} = \operatorname{prox}_{\tau \cdot h}\left(x^k - \tau \nabla f(x^k)\right)$
end for
```

The algorithm performs an explicit gradient step for f. By solving the proximal operator, an implicit step for h is performed. For $h(x) = 0$, the algorithm reduces to the basic gradient descent. The proximal gradient descent has a proven rate of convergence of $\mathcal{O}\left(\frac{1}{k}\right)$ for any step size $0 < \tau \leq \frac{1}{L}$ which is the same as for the gradient descent [49]. Hence, the proximal mapping does not have any influence on the rate of convergence. If we have $h(x) = \|x\|_{l_1}$, the algorithm is called forward backward splitting algorithm [21, 27] or Iterative Shrinkage and Thresholding Algorithm (ISTA) as proposed in [3].

3.5 Saddle-Point Problems

We first consider a general formulation of the primal problem stated in Eqn. (3.32)

$$\min_{x \in X} f(Kx) + h(x) \tag{3.33}$$

where K is a linear operator and f, h are proper, convex, lower-semicontinuous (l.s.c.) functions. We already know that this is difficult to minimize if $F(Kx)$ equals the Total Variation (TV) semi-norm $\|\nabla x\|_{TV}$ which has a discontinuity at $x = 0$. The primal-dual algorithm proposed by Chambolle and Pock [17] is suitable for these kinds of problems. The min-max saddle-point problem is defined as follows:

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + h(x) - f^*(y) \tag{3.34}$$

where X, Y are Hilbert spaces with defined scalar product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$, y is a dual variable and $f^*(y)$ is the convex conjugate of $f(x)$. The operator $K : X \mapsto Y$ is a continuous, linear, bounded operator with following properties:

$$\|K\|_{op} = \sup_{x \in X, x \neq 0} \frac{\|Kx\|_Y}{\|x\|_X} \tag{3.35}$$

$$\frac{\|Kx\|_Y}{\|x\|_X} \leq \|K\|_{op} \tag{3.36}$$

$$\|Kx\|_Y \leq L \|x\|_X \tag{3.37}$$

where L is an upper bound for $\|K\|_{op}$. The operator norm $\|K\|_{op}$ defines the smallest L for which Eqn. (3.37) holds

$$L = \|K\|_{op,2} = \sqrt{\lambda_{max}(K^T K)} \tag{3.38}$$

where λ_{max} is the largest eigenvalue of $K^T K$. The solution of the saddle-point problem can be obtained by Algorithm 5.

Algorithm 5 Primal-dual algorithm

```
Choose $x^0 \in X$, $y^0 \in Y$, $\tau, \sigma > 0$, $\tau\sigma L^2 < 1$. Set $\bar{x}^0 = x^0$, $\theta = 1$.
for $k \geq 0$ do
    $y^{k+1} = \text{prox}_{\sigma \cdot f^*}\left(y^k + \sigma K \bar{x}^k\right)$
    $x^{k+1} = \text{prox}_{\tau \cdot h}\left(x^k - \tau K^T y^{k+1}\right)$
    $\bar{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k)$
end for
```

We see that the primal-dual algorithm performs alternating gradient ascent steps for the dual and gradient descent steps for the primal variable. The step sizes τ, σ are chosen such that $\tau\sigma L^2 < 1$. According to [17] the algorithm converges if

$$\tau\sigma L^2 \leq 1. \tag{3.39}$$

The step sizes are often set to $\tau = \sigma = \frac{1}{L}$ and $\theta = 1$. Furthermore, this algorithm has a proven rate of convergence of $\mathcal{O}\left(\frac{1}{k}\right)$.

Using this information, we can now solve the variational problems presented in Chapter 2. We estimate now the primal-dual algorithm for our isotropic segmentation model. The discrete isotropic segmentation model is defined as

$$\begin{aligned} \min_u \|\nabla_g u\|_{TV} + \lambda \langle u, w\rangle \qquad &\text{s.t.} \\ u \in C_1,\, C_1 = \{u :\, u_\mathbf{i} \in [0,1],\, \forall \mathbf{i}\}& \end{aligned} \tag{3.40}$$

where $\nabla_g u$ is the discrete g-weighted TV semi-norm defined in Section 2.3, $w \in \mathbb{R}^{MND}$ is the user provided weighting function and $u \in \mathbb{R}^{MND}$ is the unknown segmentation. The image dimension is given by $M \times N \times D$. We apply the Legendre-Fenchel transform to $g_\mathbf{i}|\nabla u_\mathbf{i}|$, $\forall \mathbf{i} = (i, j, k)$. For simplicity, let $q_\mathbf{i} = (\nabla u)_\mathbf{i}$:

$$\begin{aligned} f^*(p_\mathbf{i}) &= \sup_{q_\mathbf{i}} \langle q_\mathbf{i}, p_\mathbf{i}\rangle - g_\mathbf{i}|q_\mathbf{i}| \\ &= I_{P_\mathbf{i}}(p_\mathbf{i}), \qquad P_\mathbf{i} = \{p_\mathbf{i} :\, \|p_\mathbf{i}\| \leq g_\mathbf{i},\, \forall \mathbf{i}\}. \end{aligned} \tag{3.41}$$

This is used to reformulate Eqn. (3.40) as primal-dual problem:

$$\begin{aligned} &\min_u \max_p \langle \nabla u, p\rangle + \underbrace{\lambda \langle u, w\rangle}_{h(u)} - \underbrace{I_P}_{f^*(p)} \qquad \text{s.t.} \\ &u \in C_1,\, C_1 = \{u :\, u_\mathbf{i} \in [0,1],\, \forall \mathbf{i}\} \\ \\ &\min_u \max_p \langle \nabla u, p\rangle + \lambda \langle u, w\rangle \qquad \text{s.t.} \\ &u \in C_1,\, C_1 = \{u :\, u_\mathbf{i} \in [0,1],\, \forall \mathbf{i}\}, \\ &p \in P,\, P = \{p :\, \|p_\mathbf{i}\| \leq g_\mathbf{i},\, \forall \mathbf{i}\}. \end{aligned} \tag{3.42}$$

The corresponding primal-dual algorithm for interactive image segmentation is stated in Algorithm 6.

Algorithm 6 Primal-dual algorithm for the 3D isotropic segmentation model

```
Input: $w, g_{\mathbf{i}}$
Choose $u^0 \in \mathbb{R}^{MND}$, $p^0 \in \mathbb{R}^{3MND}$ $\tau = \sigma = \frac{1}{L}$. Set $\bar{u}^0 = u^0$, $\theta = 1$.
for $k \geq 0$ do
    $\hat{p}^{k+1} = p^k + \sigma \nabla \bar{u}^k$
    $p_{\mathbf{i}}^{k+1} = \frac{\hat{p}_{\mathbf{i}}^{k+1}}{\max(1, \|\hat{p}_{\mathbf{i}}^{k+1}\| / g_{\mathbf{i}})}$
    $\hat{u}^{k+1} = u^k - \tau \nabla^T p^{k+1}$
    $u^{k+1} = \hat{u}^{k+1} - \tau \lambda w$
    $\bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k)$
end for
```

In this case, we use the nabla operator ∇ as linear operator. The operator norm $L = \|\nabla\|_{op} = \|\text{div}\|_{op}$ is estimated as:

- 2D: $L = \sqrt{4\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)}$
- 3D: $L = \sqrt{4\left(\frac{1}{h_x^2} + \frac{1}{h_y^2} + \frac{1}{h_z^2}\right)}$

where h_x, h_y, h_z denote the image spacings. For $h_x = h_y = h_z = 1$ we get:

- 2D: $L = \sqrt{8}$
- 3D: $L = \sqrt{12}$

The derivations can be found in Appendix B.

Note that the segmentation u is continuous. To achieve a binary segmentation we have to threshold u. If it is not explicitly mentioned, a threshold of 0.5 is used.

Diagonal Preconditioning

In the last section, we saw that the primal and dual step size τ, σ are estimated using the operator norm L. The constant L can be easily estimated if the operator K has a simple structure such as $K = \nabla$. For more complicated operators K, L is often very large resulting in a slow convergence of the algorithm. Therefore, we use the concept of preconditioning introduced in [52] which leads to dimension-dependent step sizes τ_j, σ_i instead of global step sizes τ, σ. Using preconditioning, we do not have to estimate the operator norm of K. Furthermore, the algorithm converges faster if K has a irregular structure. The dimension-dependent step sizes can be described using diagonal matrices $\mathbf{T} = \text{diag}\left(\tau_1, ..., \tau_n\right), \mathbf{\Sigma} = \text{diag}\left(\sigma_1, ..., \sigma_m\right)$ and are estimated as follows:

$$\tau_j = \frac{1}{\sum\limits_{i=1}^{m} |K_{i,j}|^{2-\alpha}}, \quad \sigma_i = \frac{1}{\sum\limits_{j=1}^{n} |K_{i,j}|^{\alpha}}, \quad \alpha \in [0, 2]. \tag{3.43}$$

It was shown that the algorithm converges for any $\alpha \in [0,2]$, $\theta = 1$ if

$$\|\mathbf{T}^{\frac{1}{2}} K \mathbf{\Sigma}^{\frac{1}{2}}\|^2 \leq 1 \tag{3.44}$$

which is analogue to Eqn. (3.39). Additionally, one can scale the step sizes τ_j, σ_i by a factor $\frac{1}{\xi}, \xi$, respectively, which sometimes results in better convergence. The general primal-dual algorithm including diagonal preconditioning is shown in Algorithm 7.

Algorithm 7 Primal-dual algorithm using diagonal preconditioning

```
Choose $x^0 \in X$, $y^0 \in Y$, $\mathbf{T}, \mathbf{\Sigma}$. Set $\bar{x}^0 = x^0$, $\theta = 1$.
for $k \geq 0$ do
    $y^{k+1} = \text{prox}_{\mathbf{\Sigma} f^*}\left(y^k + \mathbf{\Sigma} K \bar{x}^k\right)$
    $x^{k+1} = \text{prox}_{\mathbf{T} h}\left(x^k - \mathbf{T} K^T y^{k+1}\right)$
    $\bar{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k)$
end for
```

In this chapter, we showed basic mathematical tools to solve convex variational problems. In the next chapter, we set up a basic framework to account for different shape constraints. We introduce global and model specific shape constraints and show how we can include them into our framework.

4

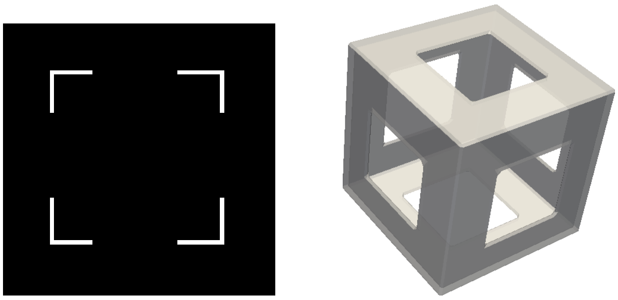
Shape Constraints

Contents

In the last chapters, we introduced a convex, Geodesic Active Contours (GAC) based image segmentation framework along with mathematical tools to solve it. Up to now, we know that we can guide the segmentation algorithm by manual interaction. In this chapter, we illustrate why segmentation solely based on user provided foreground and background scribbles is laborious, especially when dealing with volumetric images. This motivates the use of shape constraints that cover the main part of this chapter. We show how we can describe an object's shape globally. Global constraints are used to restrict for example the volume or the Center of Gravity (CoG) of a segmentation or we can force a segmentation to be (star) convex. This is especially useful for simpler shapes. If shapes appear more complex, we have to use a more specific model. We further illustrate how to incorporate a specific model into our variational segmentation framework.

4.1 User Interaction in Convex Variational Image Segmentation

In Section 2.2.4, we introduced a convex variational image segmentation framework where the user guides the segmentation process by drawing foreground and background scribbles. These scribbles are important to constrain the appearance of the final segmentation. Additionally, the framework is based on GAC, i.e., it takes image edges into account. However, image edges are not always perfect, for example if we deal with non-closed or weak image edges. In this section, we illustrate how much user interaction is really needed



(a) Input in 2D **(b)** Input in 3D

Figure 4.1: Artificial input images

for segmenting two and three dimensional data assuming imperfect edges and why this gives rise to incorporate further shape constraints.

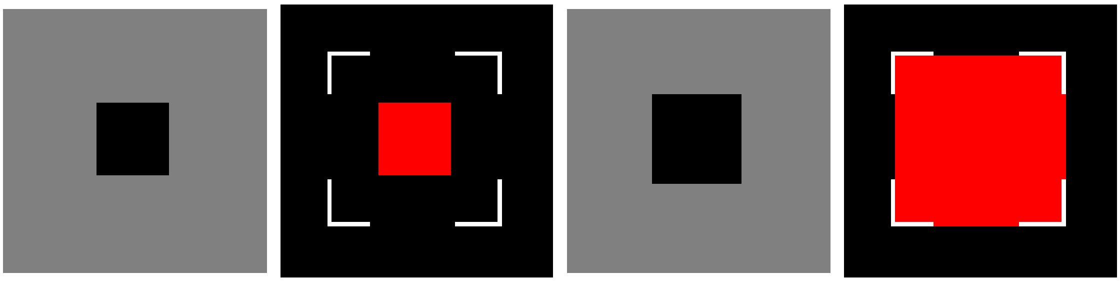
We recap the discrete isotropic segmentation model:

$$\begin{aligned} &\min_u \|\nabla_g u\|_{TV} + \lambda \langle u, w \rangle \qquad \text{s.t.} \\ &u \in C_1,\, C_1 = \{u : \, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\} \end{aligned} \tag{4.1}$$

where u is the segmentation. As input, we need edge information $g_{\mathbf{i}} \in [0,1]$ and user constraints stored in w. We consider the 2D example depicted in Fig. 4.1a. We want to segment a square with length a. It has non-closed edges which means that only a certain percentage k of the edges are visible. We paint foreground scribbles in the shape of a square of length a_{user} somewhere inside the square we want to segment. The corresponding w function is set to $w = -\infty$ where we placed foreground scribbles. The border of the image is set to background and thus, $w = +\infty$. This is illustrated in Fig. 4.2. Next, we consider image edges. The edge function $g_{\mathbf{i}}$ is zero at the borders of the square in Fig. 4.1a and one elsewhere. We estimate now how big the user's square a_{user} must be to estimate a correct segmentation, i.e., a square of length a. We require following inequality to segment the big square:

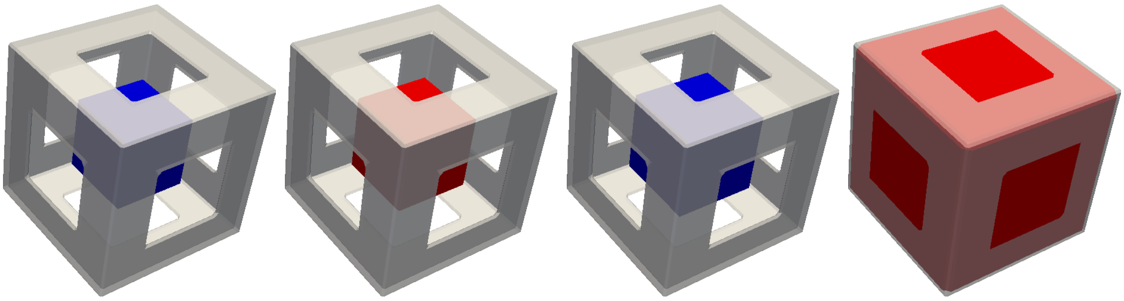
$$\begin{aligned} \|\nabla_g u\|_{TV} + \lambda \langle u, w \rangle &\leq \|\nabla_g u_{user}\|_{TV} + \lambda \langle u, w \rangle \\ (1 \cdot (1-k) + 0 \cdot k)\, 4\, a &\leq 4\, a_{user} \end{aligned} \tag{4.2}$$

where k is the percentage of visible edges. Hence, the function $g_{\mathbf{i}}$ is zero at k percent and one at $(1-k)$ percent of the edges. The left hand side denotes the energy of our desired segmentation. We assume that u is a binary function. Thus, the Total Variation (TV) semi-norm equals the perimeter of a square (see co-area formula Eqn. (2.13)). The right hand side denotes the energy of the segmentation u_{user} corresponding to the user's foreground scribbles denoting a square of length a_{user}. The edge function g equals one. The term $\lambda \langle u, w \rangle$ can be neglected as it is constant for both sides. Rearranging Eqn. (4.2)



(a) User input $a_{user} = 0.4\,a$ **(b)** Segmentation $a_{user} = 0.4\,a$ **(c)** User input $a_{user} = 0.5\,a$ **(d)** Segmentation $a_{user} = 0.5\,a$

Figure 4.2: The goal is to segment the square where 50% of the edges are missing. We show the influence of the user's input square (black) denoting a foreground scribble on the final segmentation (red) in 2D depending on the size of the input square.



(a) User input $a_{user} = 0.4\,a$ **(b)** Segmentation $a_{user} = 0.4\,a$ **(c)** User input $a_{user} = 0.5\,a$ **(d)** Segmentation $a_{user} = 0.5\,a$

Figure 4.3: The goal is to segment a cube where 25% of the surface are missing. We show the influence of the user's input cube (blue) denoting a foreground scribble on the final segmentation (red) in 3D depending on the size of the input cube.

leads to:

$$a_{user} \geq (1-k)\,a, \quad k \in (0,1). \tag{4.3}$$

The result for different values of k is depicted in Fig. 4.2. While in 2D we minimize the perimeter, we minimize the surface in 3D. Examining the same example in 3D, i.e., looking at a cube with surface $6\,a_{3D}^2$, leads to:

$$a_{user,3D} \geq \sqrt{(1-k)}\,a_{3D}, \quad k \in (0,1). \tag{4.4}$$

Figure 4.1b illustrates the input example in 3D, the corresponding results are shown in Fig. 4.3. Table 4.1 compares the results for different values k in 2D and 3D. We see that for 3D we need to draw far more foreground scribbles. Furthermore, the amount of user interaction increases when we have incomplete edges. In the following sections, we show how we can decrease the amount of user interaction and interaction time with additional shape constraints.

Table 4.1: Required fraction of length of input square (2D) and cube (3D) to length of the desired object for different percentages of visible edges k.

k	2D: $(1-k)$	3D: $\sqrt{1-k}$
0.90	0.10	0.32
0.75	0.25	0.50
0.50	0.50	0.71
0.25	0.75	0.87

4.2 Basic Segmentation Framework

Before we are going to explain different shape constraints in detail, we introduce our basic segmentation framework based on models explained in Chapter 2. We extend the segmentation model considering anisotropic edge information with an arbitrary number n of additional constraints which are modeled in terms of convex sets. Thus, we have to project the continuous segmentation $u \in \mathbb{R}^{MND}$ onto the intersection of convex sets:

$$\begin{aligned} \min_u \|D^{\frac{1}{2}}\nabla u\|_{TV} + \lambda \langle u, w\rangle \qquad \text{s.t.} \\ u \in C, \quad C = \cap_{i=1}^{n} C_i \\ C_1 = \{u : \, u_{\mathbf{i}} \in [0,1], \, \forall \mathbf{i}\} \end{aligned} \tag{4.5}$$

where $D^{\frac{1}{2}}$ is a matrix modeling the diffusion tensor for every discrete $u_{\mathbf{i}}$. We now introduce a new operator $K = D^{\frac{1}{2}}\nabla$ and a dual variable $\mathbf{p} = p$ to obtain a general model. Our primal-dual problem turns into:

$$\begin{aligned} \min_u \langle Ku, \mathbf{p}\rangle + \lambda \langle u, w\rangle \qquad \text{s.t.} \\ u \in C, \quad C = \cap_{i=1}^{n} C_i \\ C_1 = \{u : \, u_{\mathbf{i}} \in [0,1], \, \forall \mathbf{i}\} \\ p \in P, \quad P = \{p : \, \|p_{\mathbf{i}}\| \leq 1, \, \forall \mathbf{i}\}. \end{aligned} \tag{4.6}$$

We derive a general primal-dual Algorithm 8. Note that we have to perform preconditioning if K has an irregular structure (see Section 3.5). For isotropic edge handling, we set $K = \nabla$ and Line 5 of Algorithm 8 changes to the projection of a circle with radius $g_{\mathbf{i}}$. Furthermore, no preconditioning has to be performed due to the regular structure of K and global step sizes τ, σ can be used. More details can be found in Section 3.5. In the following, we focus on the more general case of anisotropic edge handling.

The projection onto the intersection of convex sets can be performed in two ways. We can either use Dykstra's projection algorithm presented in Section 3.4.2 or we can include the constraints directly in the primal-dual framework using additional Lagrange multipliers (dual variables).

4.2.1 Primal-Dual Algorithm combined with Dykstra's Projection Algorithm (PD-Dykstra)

We set $K = D^{\frac{1}{2}}\nabla$ and $\mathbf{p} = p$. In every iteration of the primal-dual algorithm, we have to project the primal variable u on the intersection of convex sets using Dykstra's projection algorithm. The initialization as well as choosing a suitable number of iterations for Dykstra's projection algorithm is crucial. Otherwise, the constraints are not fulfilled properly leading to a wrong solution.

4.2.2 Including Constraints directly in Primal-Dual Algorithm (PD)

We can also include convex constraints directly in the primal-dual algorithm using additional dual variables q. We rearrange the problem such that we obtain a new operator K and summarize all dual variables in one vector $\mathbf{p}$. Note again that we have to perform preconditioning on the new operator K and thus select a suitable parameter α and scaling factor ξ (see Section 3.5). The used notation is as follows:

$$K = \begin{bmatrix} D^{\frac{1}{2}}\nabla \\ K_{q_1} \\ \vdots \\ K_{q_n} \end{bmatrix}, \quad \mathbf{p} = \begin{pmatrix} p \\ q_1 \\ \vdots \\ q_n \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \Sigma_p & 0 & \dots & 0 \\ 0 & \Sigma_{q_1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \Sigma_{q_n} \end{bmatrix}. \tag{4.7}$$

Algorithm 8 Primal-dual algorithm for variational image segmentation with additional (shape) constraints

```
Input: $w, K$
Choose $u^0$, $\mathbf{p}^0$, $\mathbf{T}, \mathbf{\Sigma}$. Set $\bar{u}^0 = u^0$, $\theta = 1$.
for $k \geq 0$ do
    $\hat{\mathbf{p}}^{k+1} = \mathbf{p}^k + \mathbf{\Sigma} K \bar{u}^k$
    $p_{\mathbf{i}}^{k+1} = \frac{\hat{p}_{\mathbf{i}}^{k+1}}{\max(1, \|\hat{p}_{\mathbf{i}}^{k+1}\|)}$
    if $\exists\, q_i, i = 1, ..., n$ (do not use Dykstra) then
        Perform projections on dual variables $q_i$
    end if
    $\hat{u}^{k+1} = u^k - \mathbf{T} K^T \mathbf{p}^{k+1} - \lambda \mathbf{T} w$
    if Use Dykstra then
        Use Dykstra's projection algorithm to project onto the intersection of convex sets
        .
    else
        $u^{k+1} = \max\left(0, \min\left(\hat{u}^{k+1}, 1\right)\right)$
    end if
    $\bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k)$
end for
```

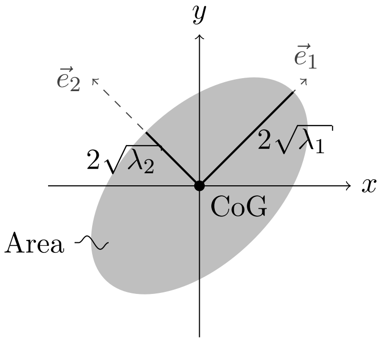


Figure 4.4: The important image moments area, CoG and covariance matrix can be extracted from the shape of an ellipse. The eigenvalues of the covariance matrix are proportional to the square of the semi-axes.

4.3 Global Shape Constraints

Making assumptions on the global shape appearance of an object can considerably increase segmentation accuracy. In this section, we focus on image moment constraints which describe properties like area, CoG and covariance, and the star prior constraint, which is useful when dealing with (star) convex objects. Global constraints are formulated in terms of convex sets. We show also how these global constraints are included in the PD-Dykstra and the PD algorithm.

4.3.1 Moment Constraints

In the following, we review the moment constraints presented by Klodt et al. [44, 45]. Especially lower order moment constraints can be useful to constrain the global shape of segmentations. The important lower-order moments are:

- 0^{th} order: Area (2D) / Volume (3D)
- 1^{st} order: Centroid / CoG
- 2^{nd} order: Covariance

As the three image moments are related to an ellipse (see Fig. 4.4), the constraints can be extracted from a user-specified ellipse (ellipsoid in 3D). We can also provide them manually.

4.3.1.1 Area / Volume Constraint

The area (and likewise the volume) constraint ensures that the area of the segmented object equals the specified area. We constrain the area a by forcing the segmentation u to lie in the convex set

$$\hat{C}_{Area} = \left\{ u : \int_\Omega u \, \mathrm{d}x = a \right\} . \tag{4.8}$$

For practical implementation, we have to work in the discrete domain and above equation turns into:

$$C_{Area} = \left\{ u : \mathbf{1}^T u = a \right\}. \tag{4.9}$$

Solution with PD-Dykstra algorithm

The projection onto the convex set C_{Area} is obtained by solving:

$$\arg\min_u \frac{1}{2} \| u - \hat{u} \|^2 \qquad \text{s.t.} \qquad \mathbf{1}^T u - a = 0. \tag{4.10}$$

To account for the equality constraint, we introduce a Lagrange multiplier $\nu \neq 0$ and get the Lagrangian function

$$\mathcal{L}(u, \nu) = \frac{1}{2} \| u - \hat{u} \|^2 + \nu (\mathbf{1}^T u - a). \tag{4.11}$$

Calculating $\nabla_u \mathcal{L}(u, \nu) = 0$ yields the stationarity condition. If we calculate the gradient with respect to ν, we obtain the equality constraint.

$$\nabla_\nu \mathcal{L}(u, \nu) : \quad \mathbf{1}^T u - a = 0 \tag{4.12}$$

$$\nabla_u \mathcal{L}(u, \nu) : \quad u - \hat{u} + \nu \mathbf{1} = 0. \tag{4.13}$$

We want to solve for the Lagrange multiplier ν:

$$\begin{aligned} u - \hat{u} + \nu \mathbf{1} &= 0 \qquad | \cdot \mathbf{1}^T \\ \underbrace{\mathbf{1}^T u}_{a} - \mathbf{1}^T \hat{u} + \nu \mathbf{1}^T \mathbf{1} &= 0 \\ a - \mathbf{1}^T \hat{u} + \nu n &= 0 \\ \nu &= \frac{\mathbf{1}^T \hat{u} - a}{n}. \end{aligned} \tag{4.14}$$

Plugging in the solution for the Lagrange multiplier ν in Eqn. (4.13) yields the final solution for u:

$$u = \hat{u} - \frac{\mathbf{1}^T \hat{u} - a}{n} \mathbf{1}. \tag{4.15}$$

Solution with PD algorithm

The constraint is included in the PD algorithm as follows:

$$\begin{gathered} \min_u \max_p \left\langle D^{\frac{1}{2}} \nabla u, p \right\rangle + \lambda \left\langle u, w \right\rangle - I_P(p) \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{ u : u_{\mathbf{i}} \in [0, 1],\, \forall \mathbf{i} \}, \\ u \in C_2,\, C_2 = \{ u : \mathbf{1}^T u - a = 0 \}, \\ P = \{ p : \| p_{\mathbf{i}} \| \leq 1,\, \forall \mathbf{i} \}. \end{gathered} \tag{4.16}$$

We introduce the dual variable $q \in \mathbb{R}$ and get:

$$\begin{aligned} &\min_u \max_{p,q} \left\langle D^{\frac{1}{2}} \nabla u, p \right\rangle + \lambda \langle u, w \rangle - I_P(p) + \left\langle \mathbf{1}^T u - a, q \right\rangle \qquad \text{s.t.} \\ &u \in C_1,\, C_1 = \{u : \, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\}, \\ &P = \{p : \, \|p_{\mathbf{i}}\| \leq 1,\, \forall \mathbf{i}\}. \end{aligned} \tag{4.17}$$

Grouping the dual variables and rearranging this equation yields:

$$\begin{aligned} &\min_u \max_{p,q} \langle Ku, \mathbf{p} \rangle - \langle a, q \rangle + \lambda \langle u, w \rangle - I_P(p), \qquad \text{s.t.} \\ &u \in C_1,\, C_1 = \{u : \, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\}, \\ &P = \{p : \, \|p_{\mathbf{i}}\| \leq 1,\, \forall \mathbf{i}\} \\ &K = \begin{bmatrix} D^{\frac{1}{2}} \nabla \\ \mathbf{1}^T \end{bmatrix},\, \mathbf{p} = \begin{pmatrix} p \\ q \end{pmatrix}. \end{aligned} \tag{4.18}$$

This requires a modification of Line 7 of Algorithm 8 to $q^{k+1} = \hat{q}^{k+1} - \Sigma_q a$.

It is often of interest that the area of an object is defined as inequality constraint, i.e., we want to define either a minimum or maximum area for an object. The inequality constraints have to have the form $f_i(x) \leq 0$ and are included in a similar way as the equality constraints by using Lagrange multipliers (dual variables). While for equality constraints the Lagrange multiplier can have either sign, it is restricted for inequality constraints to be larger or equal zero.

4.3.1.2 Center of Gravity (CoG) Constraint

Taking the CoG into account, the CoG of the segmentation u has to overlap with a provided center $\mu \in \mathbb{R}^d$. The underlying convex set is defined as:

$$\hat{C}_{CoG} = \left\{ u : \frac{\int_\Omega x u \,\mathrm{d}x}{\int_\Omega u \,\mathrm{d}x} = \mu \right\} = \left\{ u : \int_\Omega (x - \mu) u \,\mathrm{d}x = 0 \right\}. \tag{4.19}$$

To rewrite this in the discrete domain we define $X \in \mathbb{R}^{n \times d}$ holding all pixel locations $\mathbf{i}$ and $S \in \mathbb{R}^{n \times d}$ where d is the dimensionality of an input image $I \in \mathbb{R}^n$.

$$X_{CoG} = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix}, \qquad S_{CoG} = \begin{bmatrix} \mu_1 & \cdots & \mu_d \\ \mu_1 & \cdots & \mu_d \\ \vdots & \vdots & \vdots \end{bmatrix}. \tag{4.20}$$

Furthermore, let $A_{CoG} = X_{CoG} - S_{CoG}$. Discretizing Eqn. (4.20) yields:

$$C_{CoG} = \left\{ u : \, (X_{CoG} - S_{CoG})^T u = 0 \right\} = \left\{ u : \, (A_{CoG})^T u = 0 \right\}. \tag{4.21}$$

Solution with PD-Dykstra algorithm

The projection is obtained by solving:

$$\arg\min_u \frac{1}{2}\|u-\hat{u}\|^2 \qquad \text{s.t.} \qquad A^T u = 0 \tag{4.22}$$

with $A \in \mathbb{R}^{n\times d'}$, where d' is the number of rows of the matrix A. We again introduce Lagrange multipliers $\nu \in \mathbb{R}^{d'\times 1}, \nu \neq 0$ and obtain the Lagrangian function:

$$\mathcal{L}(u,\nu) = \frac{1}{2}\|u-\hat{u}\|^2 + \nu^T(A^T u). \tag{4.23}$$

Calculating gradients yields:

$$\nabla_\nu \mathcal{L}(u,\nu): \quad A^T u = 0 \tag{4.24}$$

$$\nabla_u \mathcal{L}(u,\nu): \quad u - \hat{u} + A\nu = 0. \tag{4.25}$$

The solution for the Lagrange multipliers ν is obtained as follows:

$$\begin{aligned} u - \hat{u} + A\nu &= 0 \qquad |A^T \text{ from left} \\ \underbrace{A^T u}_{0} - A^T\hat{u} + A^T A\nu &= 0 \\ A^T A\nu &= A^T\hat{u} \\ \nu &= \left(A^T A\right)^{-1} A^T\hat{u} = A^+\hat{u}. \end{aligned} \tag{4.26}$$

The expression $\left(A^T A\right)^{-1} A^T$ equals the pseudo-inverse A^+. Finally, we obtain for u:

$$u = \hat{u} - AA^+\hat{u}. \tag{4.27}$$

Solution with PD algorithm

The constraint is included in the primal-dual algorithm as follows:

$$\begin{gathered} \min_u \max_p \left\langle D^{\frac{1}{2}}\nabla u, p \right\rangle + \lambda\langle u, w\rangle - I_P(p) \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u :\, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\} \\ u \in C_2,\, C_2 = \{u :\, A^T u = 0\}, \\ P = \{p :\, \|p_{\mathbf{i}}\| \leq 1,\, \forall \mathbf{i}\}. \end{gathered} \tag{4.28}$$

We introduce a dual variable $q \in \mathbb{R}^{d'}$ which results in:

$$\begin{aligned} \min_u \max_{p,q} \langle Ku, p\rangle + \lambda \langle u, w\rangle - I_P(p) + \left\langle A^T u, q\right\rangle \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u:\, u_\mathbf{i} \in [0,1],\, \forall \mathbf{i}\}, \\ P = \{p:\, \|p_\mathbf{i}\| \leq 1,\, \forall \mathbf{i}\}. \end{aligned} \tag{4.29}$$

Rearranging this equation yields:

$$\begin{aligned} \min_u \max_{\mathbf{p}} \langle Ku, \mathbf{p}\rangle + \lambda \langle u, w\rangle - I_P(p) \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u:\, u_\mathbf{i} \in [0,1],\, \forall \mathbf{i}\}, \\ P = \{p:\, \|p_\mathbf{i}\| \leq 1,\, \forall \mathbf{i}\} \\ K = \begin{bmatrix} D^{\frac{1}{2}}\nabla \\ A^T \end{bmatrix},\, \mathbf{p} = \begin{pmatrix} p \\ q \end{pmatrix}. \end{aligned} \tag{4.30}$$

We do not have to modify Algorithm 8 because no additional projections have to be performed for the dual variable q.

4.3.1.3 Covariance Constraint

The covariance matrix Σ is a symmetric matrix relating width and height of an object. It is defined as follows:

$$\Sigma = V\Lambda V^{-1} \tag{4.31}$$

where $V \in \mathbb{R}^{d\times d}$ contains the directions of the principal axes and $\Lambda = \operatorname{diag}(\lambda_1, ..., \lambda_d)$ holds the eigenvalues of Σ such that $\lambda_1 \geq ... \geq \lambda_d$ with d being the image dimension. This describes an ellipse whose principal axes are proportional to the square root of the eigenvalues. More precisely, the length of the semi-axes a_i scale with $2\sqrt{\lambda_i}$. Thus, the covariance matrix Σ of an ellipse equals:

$$\Sigma = V \begin{bmatrix} \left(\frac{a_1}{2}\right)^2 & 0 \\ 0 & \left(\frac{a_2}{2}\right)^2 \end{bmatrix} V^{-1}. \tag{4.32}$$

The convex set to account for the dimensions of an object is defined as:

$$\begin{aligned} \hat{C}_{COV} &= \left\{u:\, \frac{\int_\Omega (x-\mu)(x-\mu)^T u \,\mathrm{d}x}{\int_\Omega u \,\mathrm{d}x} = \Sigma\right\} \\ &= \left\{u:\, \int_\Omega \left((x-\mu)(x-\mu)^T - \Sigma\right) u \,\mathrm{d}x = 0\right\}. \end{aligned} \tag{4.33}$$

Again, we have to introduce a new notation to obtain a similar structure as for the CoG constraint in Eqn. (4.21). Let us analyze the 2D case. The covariance matrix Σ has the structure:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}. \tag{4.34}$$

Setting $m_i = (x_i - \mu)$, $i = 1, ..., n$ yields the same structure $(x_i - \mu)(x_i - \mu)^T = m_i m_i^T$:

$$M_i = \begin{bmatrix} m_{i,11} & m_{i,12} \\ m_{i,21} & m_{i,22} \end{bmatrix}. \tag{4.35}$$

We now vectorize both matrices such that they have dimensionality $1 \times d^2$:

$$\Sigma^* = \begin{pmatrix} \sigma_{11} \\ \sigma_{12} \\ \sigma_{21} \\ \sigma_{22} \end{pmatrix}^T, \qquad M_i^* = \begin{pmatrix} m_{i,11} \\ m_{i,12} \\ m_{i,21} \\ m_{i,22} \end{pmatrix}^T. \tag{4.36}$$

Due to symmetry of the matrices, we can lower the dimensionality by removing the double entries:

$$\widetilde{\Sigma} = \begin{pmatrix} \sigma_{11} \\ \sigma_{12} \\ \sigma_{22} \end{pmatrix}^T, \qquad \widetilde{M_i} = \begin{pmatrix} m_{i,11} \\ m_{i,12} \\ m_{i,22} \end{pmatrix}^T. \tag{4.37}$$

To generalize this, the vectors $\widetilde{\Sigma}$, $\widetilde{M_i}$ have the reduced dimensionality $\widetilde{d} = \frac{(d+1)d}{2}$. We now define the matrices $X_{COV} \in \mathbb{R}^{n \times \widetilde{d}}$ and $S_{COV} \in \mathbb{R}^{n \times \widetilde{d}}$ such that

$$X_{COV} = \begin{bmatrix} \widetilde{M_1} \\ \widetilde{M_2} \\ \vdots \\ \widetilde{M_n} \end{bmatrix}, \qquad S_{COV} = \begin{bmatrix} \widetilde{\Sigma} \\ \widetilde{\Sigma} \\ \vdots \end{bmatrix}. \tag{4.38}$$

Furthermore, let $A_{COV} = X_{COV} - S_{COV}$. Discretizing Eqn. (4.33) yields the same structure as Eqn. (4.21):

$$C_{COV} = \left\{u : (X_{COV} - S_{COV})^T u = 0\right\} = \left\{u : (A_{COV})^T u = 0\right\}. \tag{4.39}$$

Solution with PD-Dykstra and PD algorithm

Because the covariance constraint has the same structure as the CoG constraint, the solution can be obtained the same way as shown before. The only difference is the number of rows of the matrix A.

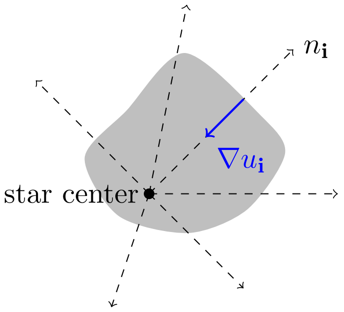


Figure 4.5: Illustration of the star shape prior. Any ray $n_{\mathbf{i}}$ sent out from the star center should be directed in opposite direction to the gradient of the segmentation $u_{\mathbf{i}}$. This constraint forces one-connected segmentations.

4.3.2 Star Shape Prior

Another global shape constraint is the star shape prior introduced by Veksler [67] and Gulshan et al. [35]. We motivate the use of the star shape prior by definition of star convexity. An object is star convex with respect to a defined star center if a straight line between any point of the object and the star center lies in the object. Examples for star convex shapes are the heart shape, star shape and any convex shape. Heart shape and star shape are only star convex for a certain amount of star centers. Convex shapes are a special case of star convex objects because they are always star convex with respect to any star center defined inside the object.

The star shape prior forces any ray sent out from a star center to be directed in opposite direction to the gradient of the segmentation u as depicted in Fig. 4.5. Thus, we define the convex set for a star shape prior as

$$C_{star} = \{u : \langle \nabla u_{\mathbf{i}}, n_{\mathbf{i}} \rangle \leq 0, \forall \mathbf{i}\} \tag{4.40}$$

with $n_{\mathbf{i}} = \begin{pmatrix} n_{\mathbf{i},x} \\ n_{\mathbf{i},y} \end{pmatrix}$ being the ray direction from the star center to the current point $\mathbf{i}$. We solve this segmentation problem using the PD algorithm:

$$\begin{gathered} \min_u \max_p \left\langle D^{\frac{1}{2}} \nabla u, p \right\rangle + \lambda \langle u, w \rangle - I_P(p) \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u : u_{\mathbf{i}} \in [0,1], \forall \mathbf{i}\}, \\ u \in C_2,\, C_2 = \{u : \langle \nabla u_{\mathbf{i}}, n_{\mathbf{i}} \rangle \leq 0, \forall \mathbf{i}\}, \\ P = \{p : \|p_{\mathbf{i}}\| \leq 1, \forall \mathbf{i}\}. \end{gathered} \tag{4.41}$$

We introduce dual variables $q \in \mathbb{R}^n$, $q_{\mathbf{i}} \geq 0$, $\forall \mathbf{i}$ to account for the inequality constraint:

$$\begin{aligned}\min_u \max_{p,q} \left\langle D^{\frac{1}{2}} \nabla u, p \right\rangle + \lambda \langle u, w \rangle - I_P(p) + \langle \langle \nabla u, n \rangle, q \rangle \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u :\, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\}, \\ P = \{p :\, \|p_{\mathbf{i}}\| \leq 1,\, \forall \mathbf{i}\}, \\ q_{\mathbf{i}} \geq 0,\, \forall \mathbf{i}.\end{aligned} \tag{4.42}$$

Rearranging this equation yields:

$$\begin{aligned}\min_u \max_p \langle K u, \mathbf{p} \rangle + \lambda \langle u, w \rangle - I_P(p) \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u :\, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\}, \\ P = \{p :\, \|p_{\mathbf{i}}\| \leq 1,\, \forall \mathbf{i}\}, \\ q_{\mathbf{i}} \geq 0,\, \forall \mathbf{i} \\ K = \begin{bmatrix} D^{\frac{1}{2}} \nabla \\ \nabla_x n_x + \nabla_y n_y \end{bmatrix},\, \mathbf{p} = \begin{pmatrix} p \\ q \end{pmatrix}.\end{aligned} \tag{4.43}$$

Due to the inequality constraint describing the star shape prior, we have to add a projection of the dual variables q. Hence, Line 7 of Algorithm 8 changes to $q_{\mathbf{i}}^{k+1} = \max(0, \hat{q}_{\mathbf{i}}^{k+1})$.

For practical implementation of the star prior we need at least one valid star center as input. It has the nice property that it forces one-connected objects. Hence, the star prior connects every foreground scribble to the star center. If a weighting map is used, the star shape prior tries to connect foreground-like regions.

Up to now, we showed only the case for selecting one star center. To segment other shapes that are not star convex, we can use an arbitrary number of star centers. In this case, we take for every pixel **i** the direction from the closest star center. Another possibility is to use every pixel of foreground scribbles as individual star centers as proposed in [35].

4.4 Model Specific Shape Constraints

In the previous section, we showed how we can describe shape appearance by global constraints. When dealing with more complex shapes, global shape constraints are not sufficient. We need to describe the complex shape by a more specific model. To generate specific shape models, we require a set of reference shapes. Statistical Shape Models (SSMs) [23] are used to describe mean shape appearance and shape variations derived from training shapes. We focus here only on the mean shape model. To obtain a mean shape, all training shapes have to be registered to a reference shape. Afterwards, these shapes are averaged which leads to a voxel-wise probability of being part of the shape. The most important question is how we can use this specific mean shape model in our segmentation framework.

Incorporating a specific shape model requires a crucial registration step. Registration can be performed manually, which is very time consuming. In [37], we proposed a registration procedure which requires only a single-point initialization to place the specific shape model in the current problem. Assuming a successfully registered shape model, we can include it in terms of an additional data term. The corresponding energy functional is:

$$\begin{gathered}\min_u \max_p \left\langle D^{\frac{1}{2}} \nabla u, p \right\rangle + \lambda \langle u, w \rangle + \lambda_{shape} \langle u, w_{shape} \rangle \qquad \text{s.t.} \\ u \in C_1,\, C_1 = \{u : \, u_{\mathbf{i}} \in [0,1],\, \forall \mathbf{i}\}, \\ P = \{p : \, \|p_{\mathbf{i}}\| \leq 1,\, \forall \mathbf{i}\}\end{gathered} \tag{4.44}$$

where w_{shape} represents the specific mean shape model and λ_{shape} a regularization parameter. To meet the requirements of a weighting map, mean shape regions are represented by negative values and background regions by a value of one. In general, we could add an arbitrary number of data terms. However, it is difficult to find a suitable set of regularization parameters.

The solution to this model is obtained by the same primal-dual Algorithm 8 as the global constraints. The additional data term is considered in the update of the primal variable u. Hence, Line 9 changes to $\hat{u}^{k+1} = u^k - \mathbf{T} K^T \mathbf{p}^{k+1} - \lambda \mathbf{T} w - \lambda_{shape} \mathbf{T} w_{shape}$.

This chapter showed the theoretical aspects of global and model specific shape constraints. In the next chapter, we evaluate the shape constraint models.

5

Evaluation of Shape Constraint Models

Contents

In this chapter, we evaluate the two presented types of shape constraints: global and model specific shape constraints. First, we give a short overview of the implementation as well as quantitative metrics to evaluate segmentations. Afterwards, we study the behavior of global shape constraints and show the applicability to real 2D and 3D data. For more complex structures, we have to use more specific shape models. We evaluate the impact of model specific constraints on the example of vertebrae segmentation.

5.1 Implementation

The algorithms for 2D are implemented in MATLAB. The user can draw an ellipse, scribbles and additional initialization points using a simple Graphical User Interface (GUI). For 3D applications, we utilized the high parallelization potential of the primal-dual algorithm. We implemented the computationally demanding algorithms using NVIDIA® CUDA to increase segmentation speed and make it usable for clinical applications. The algorithm is perfectly suited for parallel implementation because the point-wise primal and dual updates depend only on the neighboring voxels. We tested our algorithms on a PC equipped with a NVIDIA® GeForce GTX 580 graphics card (4 GB), an Intel® Core™i7 -920 processor (2.67 GHz) and 12 GB RAM.

We developed a GUI to interact with 3D medical data (see Fig. 5.1). This GUI allows to perform preprocessing (ROF or TV-l_1 denoising) and parameter selection for edge

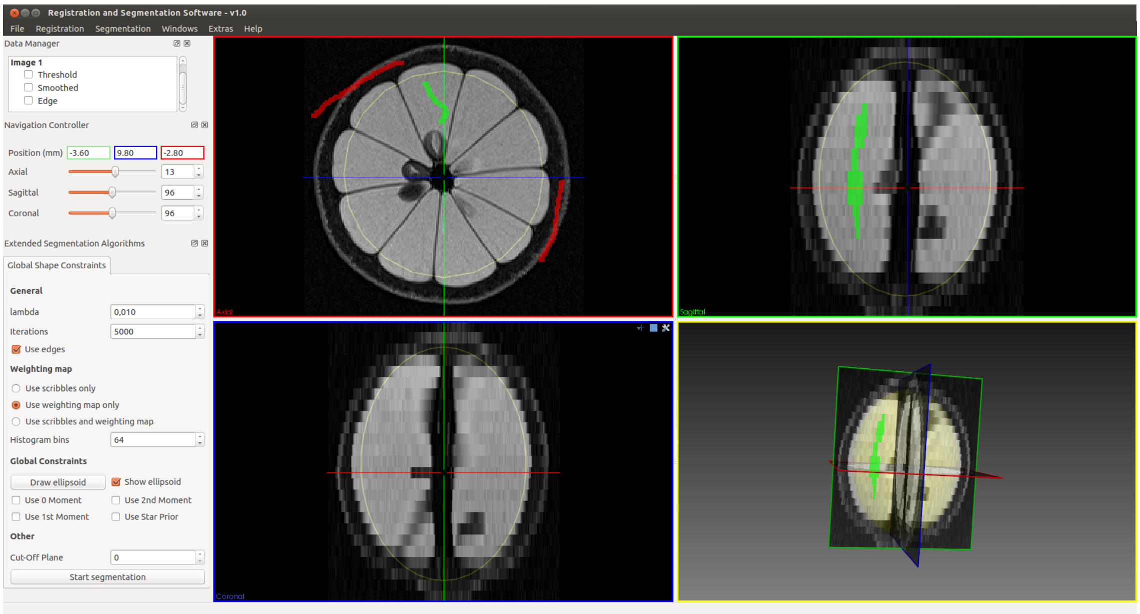


Image 1
☐ Threshold
☐ Smoothed
☐ Edge

Position (mm) -3.60 9.80 -2.80
Axial 13
Sagittal 96
Coronal 96

lambda 0,010
Iterations 5000
☑ Use edges

○ Use scribbles only
● Use weighting map only
○ Use scribbles and weighting map
Histogram bins 64

☑ Show ellipsoid
☐ Use 0 Moment
☐ Use 2nd Moment
☐ Use 1st Moment
☐ Use Star Prior

Cut-Off Plane 0

Figure 5.1: Screenshot of the GUI for segmenting volumetric medical data. The tool allows for preprocessing, registration of two data sets (which is not scope of this thesis) and Geodesic Active Contours (GAC) based segmentation with and without global shape constraints.

calculation. The shape constraints are obtained from a user-specified ellipsoid. The user can scale, translate and rotate the ellipsoid. Furthermore, the user has the possibility to draw foreground and background scribbles and to select how the weighting map should be obtained. We can select between pure local user information, obtain a weighting map from the mask given by the ellipsoid or scribbles using the log-likelihood ratio or a combination of both. The user can set the regularization parameter and the maximum number of iterations for the segmentation algorithm. After the algorithm has finished, the segmentation is displayed.

5.2 Evaluation of Image Segmentations

We evaluate the performance of our image segmentations qualitatively and if possible quantitatively. Qualitative evaluation means visual inspection of the results. In our case, we evaluate how global constraints behave in different situations. Quantitative evaluation requires reference segmentations that we can compare to. A number of evaluation metrics exist to compare a binary segmentation A against a reference (ground truth) segmentation B and evaluate the performance objectively. This also allows for comparison between different algorithms. However, especially in medicine we often do not have large, publicly available databases with reference segmentations. Different methods are evaluated on different data sets, so results are also not fully comparable either. In the following, we review commonly used quantitative metrics.

Dice Similarity Coefficient (DSC)

A widely used spatial similarity measure is the Dice Similarity Coefficient (DSC) [28]. It emphasizes the overlap between segmentations and is expressed as a percentage value

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \times 100. \tag{5.1}$$

A maximum overlap is given if the DSC is 100%. However, if we compare objects with small surface-to-volume ratio, the DSC is not very meaningful due to similar scores for good and wrong segmentations.

Mean Absolute Surface Distance (MAD)

The Mean Absolute Surface Distance (MAD) measures the average absolute euclidean distance $d(\cdot,\cdot)$ in mm from the surface S_A to the reference surface S_B as follows:

$$MAD = \frac{1}{|S_A|} \sum_{i=1}^{|S_A|} |d_i(S_A, S_B)|. \tag{5.2}$$

Hausdorff Distance (HD)

The Hausdorff Distance (HD) [38] measures how close two surfaces are. It is defined as the maximum between two directed HDs in mm:

$$HD_{max} = \max\left(HD_{A\to B}, HD_{B\to A}\right). \tag{5.3}$$

The directed HD from $A \to B$ is given as

$$HD_{A\to B} = \max_{s_A \in S_A} \min_{s_B \in S_B} d(s_A, s_B) \tag{5.4}$$

where S_A, S_B denote the set of surface voxels s_A, s_B. To neglect outliers in the distance score, the HD_{95} is used which is the Hausdorff distance for the 95th percentile.

5.3 Global Shape Constraints

In this section, we study the impact of global shape constraints and combinations of global shape constraints. We point out the difference between the two proposed methods PD-Dykstra and PD for considering additional constraints. Furthermore, we apply shape constraints to real 2D images. Finally, we show the application of global shape constraints to volumetric brain tumor segmentation and sinus floor augmentation segmentation.

5.3.1 Impact of Global Shape Constraints

We incorporate global constraints in our framework to study the advantages and disadvantages of the presented global constraints. We do not want to achieve perfect results, but to see how various global constraints and combination of global constraints behave.

5.3.1.1 Experimental Setup

The goal for this experiment is to study the behavior of various shape constraints for brain tumor segmentation. Therefore, we use a contrast enhanced, T_1 weighted Magnetic Resonance Imaging (MRI) axial brain slice taken from the BRATS data set [47]. The image is ROF denoised. Edges are calculated setting the parameters to $\alpha = 20$ and $\beta = 0.55$. All constraints are extracted from a user-specified ellipse. This ellipse is also taken to calculate a weighting map using the log-likelihood ratio, where values inside the ellipse correspond to foreground and values outside the ellipse to background regions. The regularization parameter λ is set to 0.03. We consider isotropic edges in our segmentation algorithm. As a reference segmentation of the tumor core is provided by the dataset, we compare our results quantitatively in terms of the DSC to the reference segmentation as well as qualitatively.

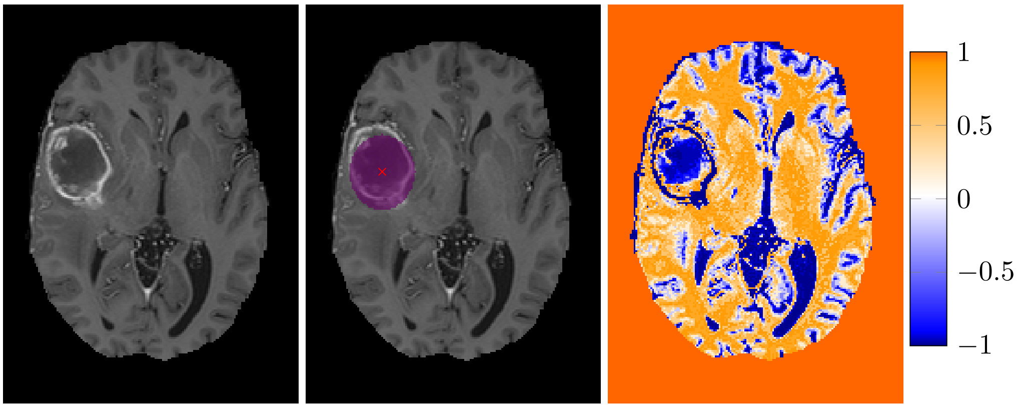
5.3.1.2 Results

The input image for this experiment is depicted in Fig. 5.2a. The user-specified ellipse (Fig. 5.2b) is used to obtain a weighting map (Fig. 5.2c) which is illustrated as color map. Orange regions depict background, blueish regions depict foreground. Bright or white regions are classified neither as foreground nor background, hence, the values in the weighting map are close to zero. The segmentations are compared to a reference segmentation (see Fig. 5.3). In the comparison, green regions denote correctly segmented areas. Red regions are wrongly segmented by our algorithm. If our segmentation algorithm does not capture the tumor core region, it is illustrated in yellow.

Before we consider shape constraints, we illustrate the result of our segmentation algorithm using the weighting map as only input in Fig. 5.3a. We achieve a clear over-segmentation. Many regions are estimated as foreground regions in the weighting map which guides our segmentation algorithm. In the following, we show how we can reduce the solution to only segment the tumor core.

Figure 5.3b depicts the result using the area constraint. The area is reduced significantly compared to the segmentation without shape constraints. It is distributed among the image. Although regions inside the tumor are classified as foreground, the algorithm favors regions with highly negative foreground values, i.e., dark blue regions, in the weighting map like the boundary of the tumor and other structures.

The CoG constraint illustrated in Fig. 5.3c is responsible for finding a segmentation u whose CoG equals the specified center of the ellipse. This results in a partly segmented



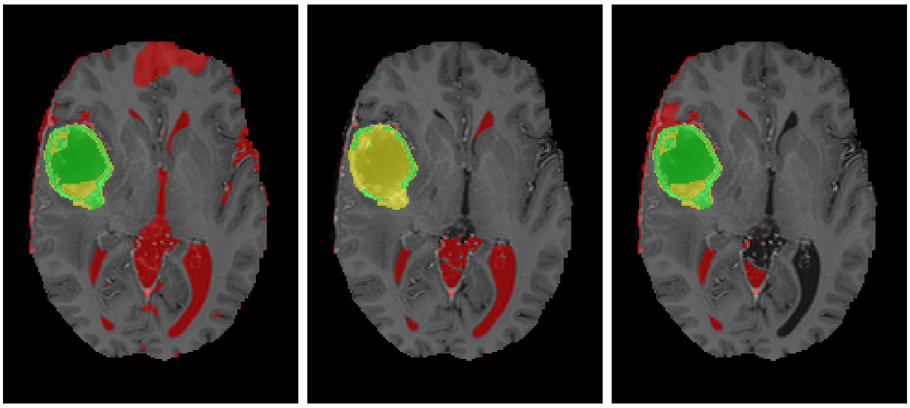
(a) Input **(b)** User-specified ellipse **(c)** Weighting map

Figure 5.2: Figure (a) shows the input image. Important information to formulate shape constraints is extracted from the ellipse in (b). The Center of Gravity (CoG) is marked by a red cross. The weighting map obtained from the user-specified ellipse is illustrated in (c). Blue depicts foreground regions, orange background regions. White regions are classified neither as foreground nor as background.

tumor. However, as the area is not restricted, many other parts are segmented along with the brain tumor. Combining area and CoG constraint further reduces the amount of spurious parts as illustrated in Fig. 5.3d. Holes remain in the tumor region. The border of the brain is still emphasized.

The covariance constraint takes the ellipse's geometry into account. It reduces further the amount of spurious parts as seen in Fig. 5.3e. The algorithm optimizes always the dimensions of the object as well as the CoG simultaneously because the CoG is fixed for this constraint (see Section 4.3.1.3). Using the covariance, the algorithm often shrinks the segmentation. We can add the area constraint which additionally fills the hole in the tumor. However, all spurious parts are not removed and a small part of the tumor is still missing. This is illustrated in Fig. 5.3f. Using only moment constraints, the combination of covariance and area constraint results in the best segmentation.

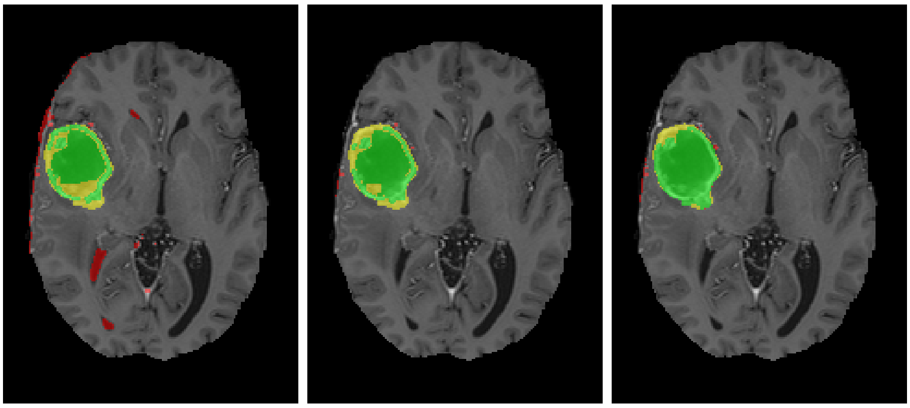
After showing the impact of image moment constraints, we focus now on the star prior constraint which is suitable for one-connected segmentations. Figure 5.3g shows the impact of the star prior. The tumor is already well segmented. Nevertheless, the star prior tries also to connect other foreground-like regions. In the worst case, this will connect to structures that are at a completely different location. This is a motivation to incorporate additional image moment constraints. Indeed, adding area constraint or CoG constraint yields a correct segmentation of the brain tumor as illustrated in Fig. 5.3h and Fig. 5.3i. In this example, combining star-prior with CoG constraint yields the best segmentation. The DSCs which are depicted along with the segmentation results in Fig. 5.3 verify our observations.



(a) No shape constraints DSC=41.50%

(b) Area DSC=13.41%

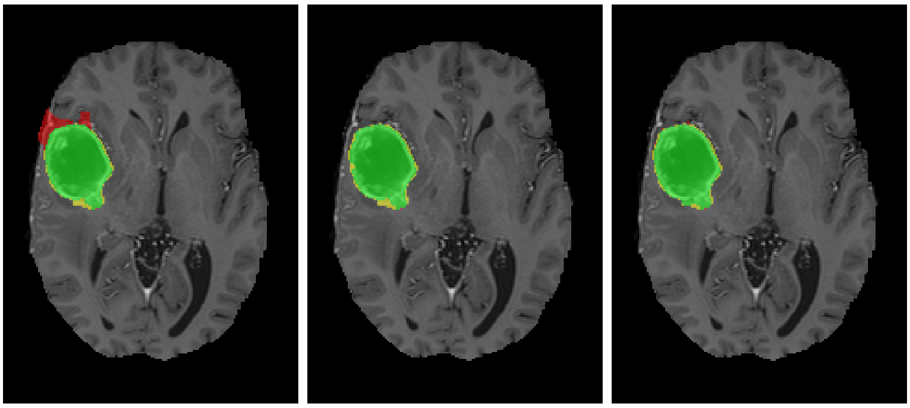
(c) CoG DSC=68.76%



(d) Area + CoG DSC=72.96%

(e) Covariance DSC=84.45%

(f) Covariance + Area DSC=92.50%



(g) Star Prior DSC=89.48%

(h) Star Prior + Area DSC=95.51%

(i) Star Prior + CoG DSC=96.59%

Figure 5.3: All segmentations are compared to a reference segmentation qualitatively and quantitatively in terms of the DSC. Correctly segmented regions are depicted in green. Red regions are wrongly segmented by our algorithm. Yellow regions are missing in our segmentation. The segmentation using the weighting map and no further shape constraints is illustrated in (a). Impact of global shape constraints and combinations of constraint are illustrated in (b)-(i). The star center equals the CoG.

5.3.2 Comparison of PD-Dykstra and PD

In Section 4.2, we introduced two algorithms namely PD-Dykstra and PD to account for additional constraints. For PD-Dykstra, it is crucial to perform sufficient iterations such that all constraints are fulfilled. This becomes more important for complicated projections. Runtime increases linearly with the number of Dykstra iterations. In case of projection onto the CoG or covariance constraints, we have to compute a pseudo-inverse which is computationally demanding.

The advantage of PD compared to PD-Dykstra is that the PD algorithm does not require to compute any matrix inverse which reduces runtime significantly. Furthermore, we do not have to perform an extra optimization for the projection onto the intersection of convex sets because they are directly handled in the primal-dual problem. As we have an irregular structure for the linear operator, preconditioning has to be performed. We have to select parameters α and ξ for optimal convergence.

We show the dependence of our algorithm on preconditioning for the example of brain tumor segmentation with star prior and CoG constraint illustrated in Fig. 5.3i. To show the convergence of our algorithm, we first estimate an optimal solution by performing a large number of iterations (e.g. iter=10^9). We calculate the energy for the current iterate as proposed by [61]:

$$E(u^n) = \|\nabla_g u^n\|_{TV} + \lambda \langle u, w\rangle + \delta \| \frac{A^T u^n}{\sum_{\mathbf{i}} u^n_{\mathbf{i}}} \| \tag{5.5}$$

where $\frac{A^T u^n}{\sum_{\mathbf{i}} u^n_{\mathbf{i}}}$ describes the CoG constraint (see Section 4.3.1.2) and $\delta > 0$ is a suitable parameter. Let us denote the optimal energy as $E(u^*)$. The convergence is shown by plotting the difference of the current energy to the optimal energy:

$$E_{diff} = E(u^n) - E(u^*). \tag{5.6}$$

In Fig. 5.4, we illustrate the convergence of our algorithm for different values $\alpha \in (0, 2)$ and $\xi = 1$. The parameter δ is set to 10. We get faster convergence for increasing α. In the first iterates, the algorithm tries to optimize the overall energy, the segmentation u gets its rough shape. Afterwards, a plateau can be noticed in the energy plot. The last part shows a decrease in energy corresponding to fine tuning of the constraints which mainly effects single voxels. Note again that the segmentation u is continuous. The final segmentation is achieved by thresholding.

For our constraints, experiments showed that a setting $\alpha = 2$ and $\xi = 1$ yields good convergence. We observed no significant improvement for other scaling factors ξ. However, we cannot generalize this setup for other applications because the selection of α, ξ, depends on the structure of the linear operator K. In general, the factor ξ should not be too low because this lowers the dual step sizes σ_i too much to get proper convergence. If the factor ξ is too high, primal step sizes τ_j may be to low to get proper convergence.

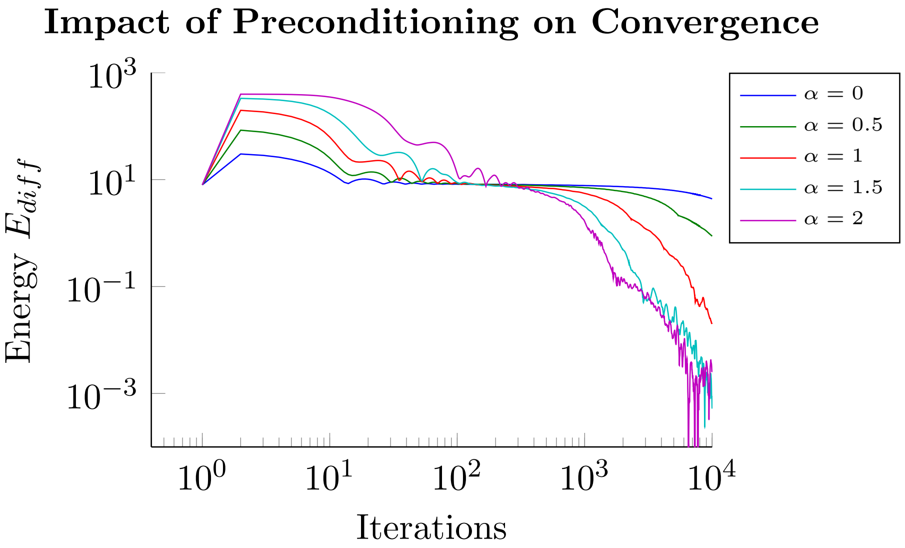


Figure 5.4: Impact of preconditioning on convergence for different values α. We see that the algorithm converges best for $\alpha = 2$.

5.3.3 Examples in 2D

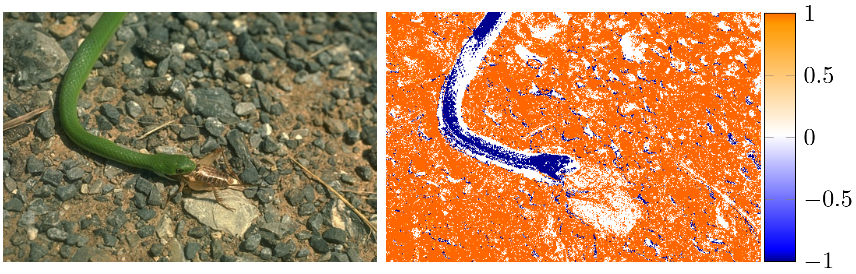
We present examples in 2D to show the great variability of global constraints. Especially when we combine constraints they show nice properties as we have already seen before. In this section, we explore the star prior on color images in combination with image moment constraints and extend it to multiple star centers. Furthermore, we compare the amount of needed user interaction with the approach that uses foreground and background scribbles only.

5.3.3.1 Experimental Setup

All input images are ROF denoised to get clearer edges. For edge calculation, the parameters are set to $\alpha = 20$ and $\beta = 0.55$. For segmentation, we considered isotropic edges.

Snake

The weighting map is calculated using the log-likelihood ratio from foreground and background scribbles. The regularization parameter is set to $\lambda = 0.005$. We compare the use of no, one and two star centers.



(a) Input image **(b)** Weighting map

Figure 5.5: Figure (a) shows the input image. The weighting map obtained from user scribbles is illustrated in (b).

Banana

The weighting map is calculated using the log-likelihood ratio from the user-defined ellipse. The regularization parameter is set to $\lambda = 0.02$. We used several star centers as input because the banana is not star convex. With this example, we show the influence of the CoG on the segmentation.

Owl

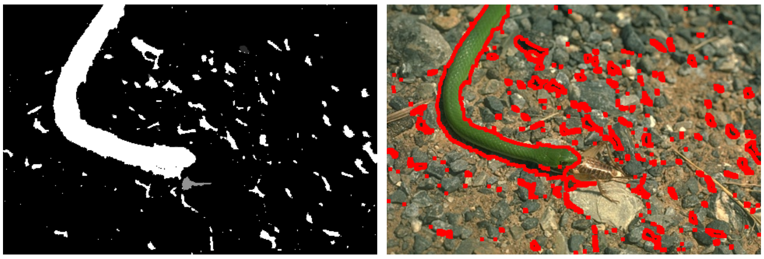
The weighting map is calculated using the generic approach of Donoser et al. [29] using the user-defined ellipse as mask. The regularization parameter is set to $\lambda = 0.02$. We illustrate how we can separate a single owl from the other owls.

5.3.3.2 Qualitative Results

In the following, we illustrate the results for the described examples. The weighting maps are shown as color maps where orange regions depict background, blue regions depict foreground and white or bright regions are classified neither as foreground nor background. Finally, we compare the required user interaction for shape constraints to interaction based on pure user scribbles.

Snake

Figure 5.5a shows the image of a snake, Fig. 5.5b illustrates the weighting map obtained from foreground and background scribbles shown in Fig. 5.7a. Using only the weighting map and no shape constraints results in a clear over-segmentation of other structures than the snake as illustrated in Fig. 5.6. Using one star center as shape constraint is not enough to describe the entire shape of the snake. Background regions are captured by the star center to connect to the upper parts of the snake (see Fig. 5.7). Using a second star center as illustrated in Fig. 5.8 segments the outline of the snake correctly.



(a) Segmentation u **(b)** Final Segmentation

Figure 5.6: Segmentation without shape constraints.

(a) Constraints **(b)** Segmentation u **(c)** Segmentation overlay

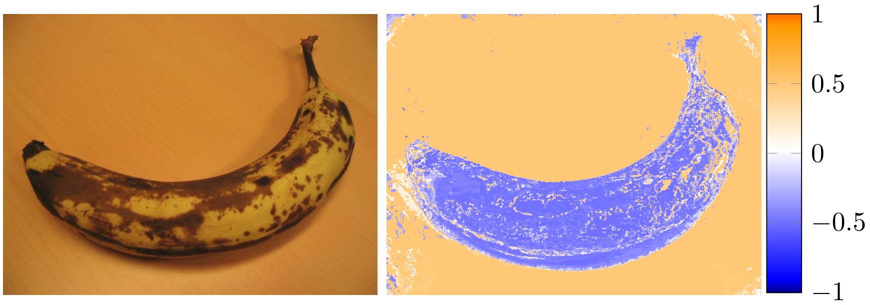
Figure 5.7: Segmentation using a single star center (green circle). The shape of the snake is not fully described by one star center. Background regions are captured when connecting to the upper part of the snake.

(a) Constraints **(b)** Segmentation u **(c)** Segmentation overlay

Figure 5.8: Segmentation using multiple star centers (green circles). Two star centers describe the shape of the snake correctely.

Banana

Using only the weighting map (Fig. 5.9b) for segmentation of a banana (Fig. 5.9a) yields an under-segmented banana as illustrated in Fig. 5.10. The shadow is segmented, some other parts of the banana are missing. Figure 5.11 shows the result when using multiple star centers. The corresponding segmentation contains the banana's shadow on the bottom. In contrast, putting a CoG yields a segmentation of the banana only as illustrated in Fig. 5.12. The shadow is not segmented in this case.



(a) Input image **(b)** Weighting map

Figure 5.9: Figure (a) shows the input image. The weighting map obtained from an ellipse input is illustrated in (b).

(a) Segmentation u **(b)** Final Segmentation

Figure 5.10: Segmentation without shape constraints.

(a) Constraints **(b)** Segmentation u **(c)** Segmentation overlay

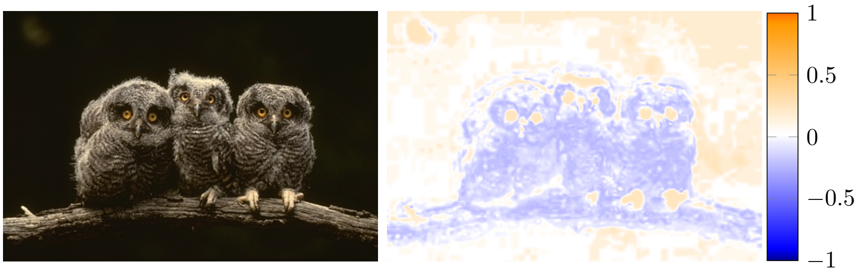
Figure 5.11: Segmentation containing the shadow of the banana. The shape constraints are multiple star centers illustrated as green circles.

Owl

Figure 5.13a shows the input image. The used weighting map is depicted in Fig. 5.13b. The segmentation without any shape constraints is illustrated in Fig. 5.14. With this example, we illustrate how to segment an owl out of a group of owls using a combination of area, covariance and star prior constraint (see Fig. 5.15). The star prior has the advantage that it closes holes such as the owl's eyes which are classified as background in our weighting

(a) Constraints **(b)** Segmentation u **(c)** Segmentation overlay

Figure 5.12: The shape constraints are multiple star centers (green circles) and a manually set CoG (red cross). The banana is segmented without the shadow.



(a) Input image **(b)** Weighting map

Figure 5.13: Figure (a) shows the input image. The weighting map obtained from an ellipse input is illustrated in (b).

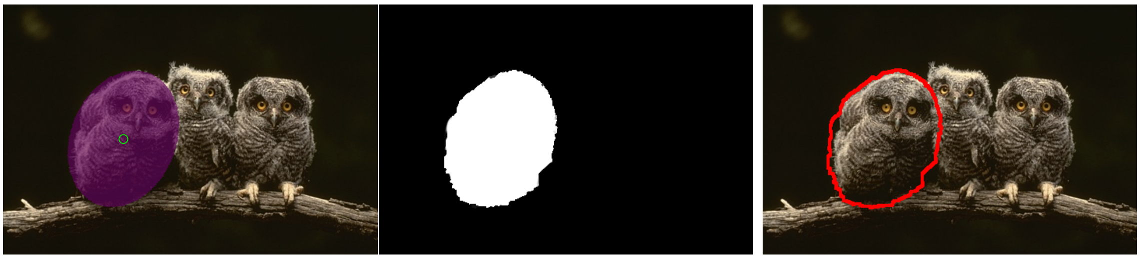
(a) Segmentation u **(b)** Final Segmentation

Figure 5.14: Segmentation without shape constraints.

map. Using the CoG combined with the area constraint is not suitable here because the area is still distributed to the branch and other owls. Using the covariance constraint keeps the proportions of the object.

Comparison with pure user scribbles

Figure 5.16 compares the user interaction for our shape constraints to pure foreground / background scribbles segmentation based on GAC. We see that we need more effort when segmenting with user scribbles, especially when dealing with highly textured objects. We



(a) Constraints **(b)** Segmentation u **(c)** Segmentation overlay

Figure 5.15: The shape constraints as well as the weighting map are obtained from a single user-specified ellipse illustrated in (a). Segmenting a single owl out of a group of owls is achieved by using covariance, area and star prior constraint. The star center equals the CoG of the ellipse and is denoted by a green circle in (a).

Figure 5.16: Comparison of user interaction to achieve the same segmentation. The top row shows manual foreground and background scribbles. Using shape constraints, the user interaction reduces to point initializations and drawing of an ellipse as depicted in the bottom row. We can also combine user scribbles with shape constraints as showed by the example of segmenting the snake.

cannot smooth the boundaries away by using a small λ in ROF denoising because we would loose important edges. Shape constraints are based on a weighting map and single point initializations of the constraints. We can combine shape constraints also with user scribbles as shown for the snake in the first column.

5.3.4 Brain Tumor Segmentation in 3D

In this section, we test our global shape constraint on real 3D examples. The aim is to segment the tumor core on contrast enhanced T_1 weighted MRI images on data sets from the BRATS 2013 database [47]. The tumor core includes enhanced regions as well as necrotic tissue inside the enhanced regions. Furthermore, post-operative cavities are also part of the tumor core region. The image spacing of the input images is $1 \times 1 \times 1$ mm.

5.3.4.1 Experimental Setup

We perform slight pre-smoothing using the ROF denoising model. The edges are calculated using the parameters $\alpha = 20$ and $\beta = 0.55$. The results are compared to the reference segmentations. We performed the experiments on two data set with different tumor sizes and compositions. Data set 1 shows enhanced tumor rims, the inside of the tumor is necrotic tissue. Data set 2 shows mainly the enhanced tumor. The weighting map for segmentation is obtained using the log-likelihood ratio based on the user-defined ellipse. The regularization parameter λ is set to 0.01 and we consider isotropic edges for segmentation. For both data sets, we use a combination of star prior, CoG and volume constraint.

5.3.4.2 Qualitative Results

Figure 5.17 illustrates the segmentation as well as the user-defined ellipse which is the only input for our shape constraints in 3D. The star center equals the CoG of the ellipsoid. Figure 5.18 depicts the segmentation result for data set 1, Fig. 5.19 shows the same for data set 2. In both cases, we do not see large deviations from the ground truth data.

5.3.4.3 Quantitative Results

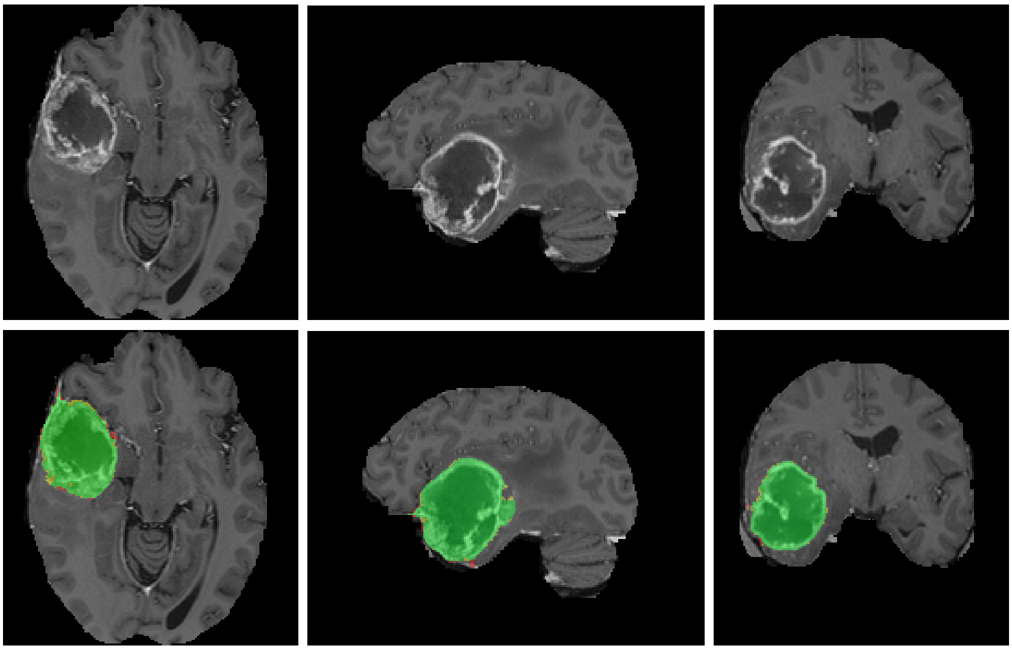
The brain tumor segmentations are evaluated quantitatively in terms of the DSC, MAD, HD_{95} and HD_{max}. Furthermore, the volume of the segmentation is calculated. The results are depicted in Table 5.1. The MAD is very low for both cases. Data set 1 has a higher DSC of 96.83% compared to data set 2 (DSC$= 94.58\%$). The HD_{max} is better for data set 2 than for data set 1. The same holds for HD_{95}. These examples show clearly that there should be a distance metric considered beside the DSC. The DSC is not very meaningful for a low surface-to-volume ratio. The surface-to-volume ratio is much lower for data set 1 than for data set 2. Furthermore it shows the difference of using HD_{95} to HD_{max}. While HD_{max} defines the largest distance between two surfaces, HD_{95} neglects outliers.

Table 5.1: Quantitative results for brain tumor core segmentation on two example data sets.

Metric	Data set 1	Data set 2
Volume in *ml*	57.53	4.71
DSC in %	96.83	94.58
MAD in *mm*	0.25	0.24
HD_{95} in *mm*	1.41	1
HD_{max} in *mm*	6.40	3.16

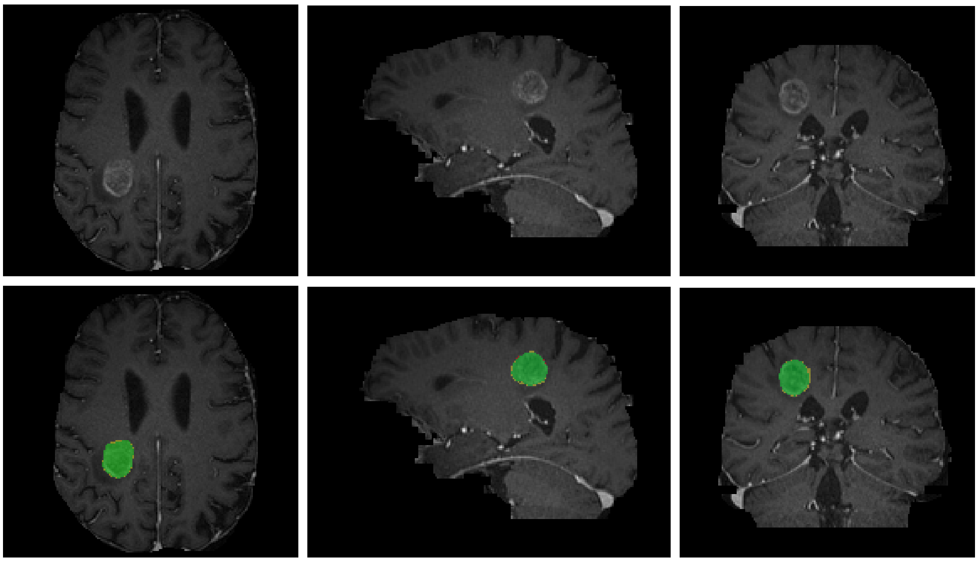
(a) Data set 1 **(b)** Data set 2

Figure 5.17: Visualization in 3D of brain tumor segmentations and user-defined ellipsoids (violet) for two example data sets.



(a) axial **(b)** sagittal **(c)** coronal

Figure 5.18: Brain tumor segmentation on data set 1. The top row depicts cross-sections from the input images. The segmentation results are illustrated in the bottom row. The segmentation is compared to the reference segmentation. Green regions denote a correct segmentation, red regions depict over-segmented areas and yellow regions a missing segmentation.



(a) axial (b) sagittal (c) coronal

Figure 5.19: Brain tumor segmentation on data set 2. The top row depicts cross-sections from the input images. The segmentation results are illustrated in the bottom row. The segmentation is compared to the reference segmentation. Green regions denote a correct segmentation, red regions depict over-segmented areas and yellow regions a missing segmentation.

5.3.5 Sinus Floor Augmentation Segmentation in 3D

The current method for grafting material segmentation in the maxillary sinus is carried out manually in clinical applications [43]. In every axial slice of the Computed Tomography (CT) volume, the boundary of the object has to be drawn. Since this method is time consuming and the whole object appearance in 3D is not regarded which leads to errors in the segmentation, one seeks for interactive methods guiding the user through this process. Using GAC based image segmentation has the advantage that the exact boundary has not to be drawn. Far less user interaction is needed than for manual segmentation. Pure weighting maps are not useful for this task because the grafting material has the same or similar intensities as bone tissue leading to over-segmentation. Streaking artifacts that appear due to the high Hounsfield Units (HU) of implants can influence our segmentation, too. We show how global shape constraints can support this ambiguous task of sinus floor augmentation segmentation.

The segmentations are prone to leak out into other teeth, implants or other parts of the jaw bone. Especially implants depicted in Fig. 5.22 are often classified as foreground. Implants are set in grafting material. Therefore, grafting material has to be cut out in order to set implants. Thus, only those parts of the implants where grafting material was cut out belongs to the final segmentations. Using the covariance constraint does not prevent this problem. We overcome this problem by defining a cutting plane in the GUI. All regions below the cutting plane are set to background in the weighting image.

5.3.5.1 Experimental Setup

We perform pre-smoothing using the ROF denoising model to obtain clear edges. The edges are calculated using the parameters $\alpha = 15$ and $\beta = 0.55$. We estimate a weighting map determined from the user-specified ellipsoid using the log-likelihood ratio. The regularization parameter is set to $\lambda = 0.02$. We experiment with two different data sets showing different compositions of grafting material. The grafting material is segmented separately in the left and right maxillary sinus. For data set 2, we defined a cutting plane to prevent the segmentation to leak out to lower parts of implants. The image spacing is for data set 1 $0.27.27 \times 1$ and for data set 2 $0.37.37 \times 1$.

5.3.5.2 Qualitative Results

We compare our results against manually annotated segmentations. Figure 5.20 and Fig. 5.22 show the results for different grafting materials segmented in the left and right maxillary sinus. Mucous membrane is present in the sinus in both cases which makes it more difficult to identify the grafting material. As grafting materials are assumed to be convex and thus one-connected, the star shape prior is suitable for this segmentation problem. Constraining the star prior further with volume and CoG constraint led to promising results. A 3D visualization of the used constraints and a comparison to using pure user scribbles is illustrated in Fig. 5.21 for data set 1 and in Fig. 5.23 for data set 2.

5.3.5.3 Quantitative Results

We test the experiment for different combinations of global constraints. The evaluation metrics for the combination of star prior, CoG and volume constraint are depicted in Table 5.2, for the combination of star prior and covariance constraint in Table 5.3. The first settings performs considerably better on both data sets. The second setting, i.e., star prior with covariance constraint performs slightly better for the left grafting material of data set 2. Combining covariance, volume and star prior constraint did not show any significant difference compared to the combination of covariance and star prior constraint.

We achieve good results for data set 1 in terms of all metrics. The comparison of HD_{95} and HD_{max} indicates that there are no large outliers. Furthermore, the low values of the HDs show that the segmentation did not leak out to other parts of the jaw bone or other teeth. Compared to data set 2, the grafting material in data set 1 differs from the jaw bone.

The results for data set 2 are similar to data set 1. The HD_{max} for the left grafting material is rather large. The segmentation leaks out compared to the reference segmentation. However, even for experts it is not easy to identify if the over-segmented part in our segmentation is really jaw bone or still grafting material.

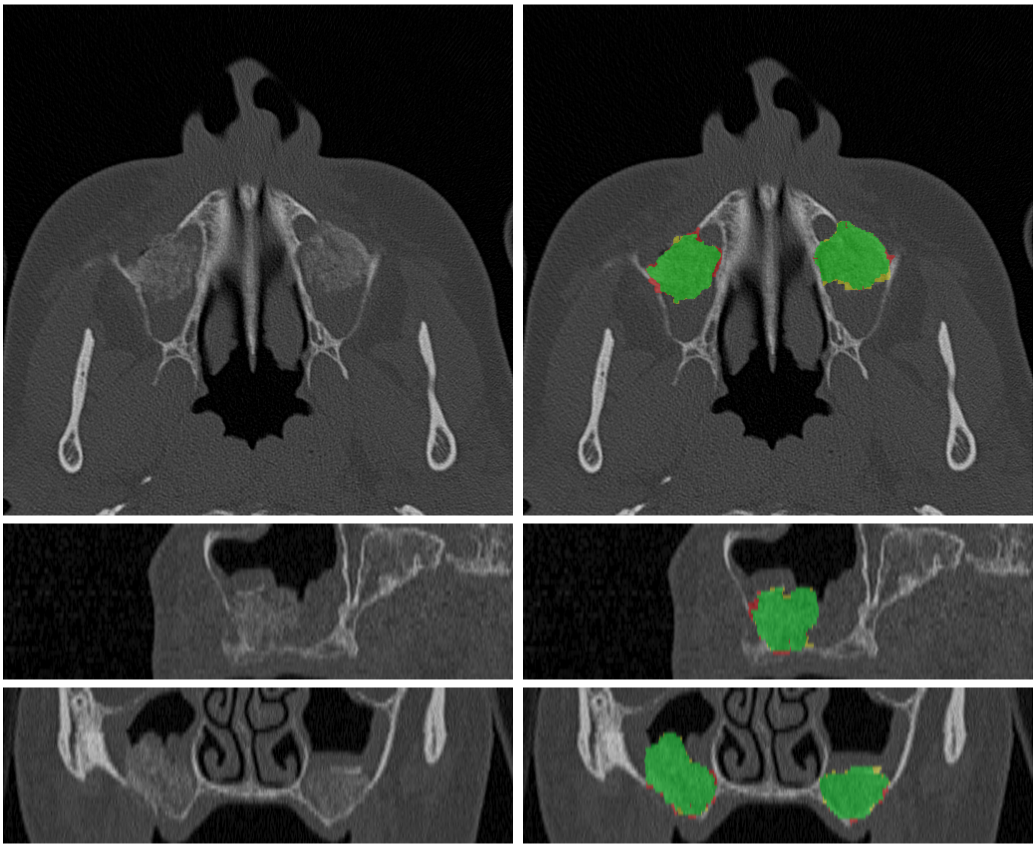
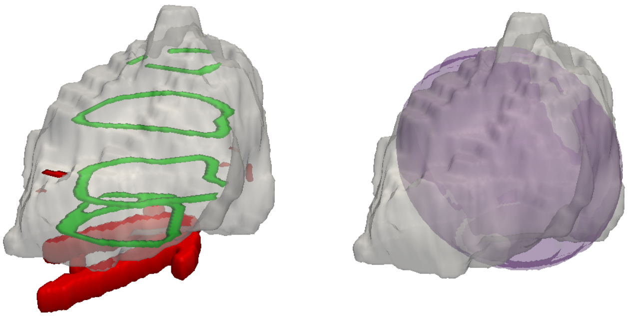


Figure 5.20: Data set 1: Slices through all image planes. We segment the grafting material in the left and right maxillary sinus. The grafting material is surrounded by mucous tissue. The edges are not clearly visible. The second column depicts the segmentation when using a combination of global shape constraints, i.e., star prior, CoG and volume constraint. We compared the segmentation to a reference segmentation. Green areas depict correctly segmented regions, red areas are over-segmented by our algorithm and yellow areas are missing.



(a) Pure user scribbles **(b)** Global shape constraints

Figure 5.21: Data set 1: Amount of user input for segmentation using solely user scribbles (a) and segmentation using star prior, CoG and volume constraint obtained from a user-specified ellipsoid (violet) (b).

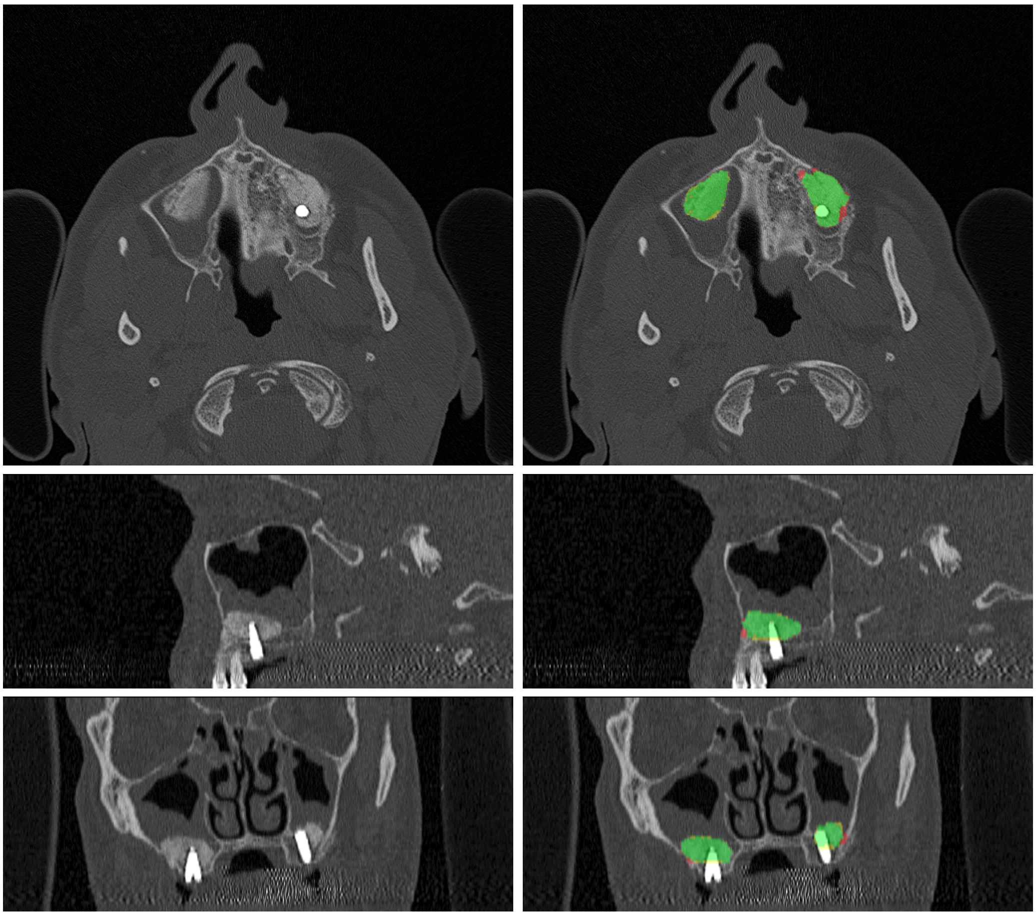
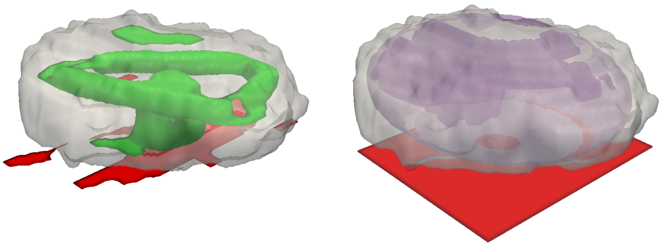


Figure 5.22: Data set 2: Slices through all image planes. We segment the grafting material in the left and right maxillary sinus. The grafting material intensities are the same as surrounding bone structures. Furthermore, the implants influcence the segmentation. Therefore, we use not only global shape constraints (star prior, volume and CoG), but specify a cutting plane to not allow to segment the whole implant. We compared the segmentation to a reference segmentation. Green areas depict correctly segmented regions, red areas are over-segmented by our algorithm and yellow areas are missing.



(a) Pure user scribbles **(b)** Global shape constraints

Figure 5.23: Data set 2: Amount of user input for segmentation using solely user scribbles (a) and segmentation using star prior, volume and CoG constraint obtained from an ellipsoid (violet) as well as a user-specified cutting plane (b).

Table 5.2: Quantitative results for sinus floor augmentation segmentation on two example data sets. We use star prior, CoG and volume constraint.

Metric	Data set 1 (left)	Data set 1 (right)	Data set 2 (left)	Data set 2 (right)
Volume in ml	3.54	2.77	2.54	1.37
DSC in %	92.41	94.20	92.53	91.49
MAD in mm	0.22	0.19	0.22	0.24
HD_{95} in mm	1.31	1.00	1.07	1.13
HD_{max} in mm	2.93	3.12	16.87	3

Table 5.3: Quantitative results for sinus floor augmentation segmentation on two example data sets. We use star prior and covariance constraint.

Metric	Data set 1 (left)	Data set 1 (right)	Data set 2 (left)	Data set 2 (right)
Volume in ml	3.18	2.62	2.73	1.89
DSC in %	87.56	90.73	93.92	87.20
MAD in mm	0.50	0.35	0.15	0.24
HD_{95} in mm	2	1.46	1.07	2.30
HD_{max} in mm	4.23	2.73	16.87	4.18

5.4 Model Specific Shape Constraints

In this section, we evaluate model specific shape constraints for vertebrae segmentation. Vertebrae segmentation is a suitable task due to the complex appearance of vertebral substructures. Furthermore, training data is available which allows to build an application specific shape model and quantitative evaluation.

5.4.1 Experimental Setup

Our vertebral shape model is derived from the CSI MICCAI challenge training data set [73] containing ten CT scans of the spine and labeled thoracic and lumbar vertebrae. As we do not have a large amount of training data available and the variation in shape along the spine varies greatly, we estimate a shape model for three groups of vertebrae:

- Upper thoracic vertebrae: T1-T6
- Lower thoracic vertebrae: T7-T12
- Lumbar vertebrae: L1-L5

The user initializes each vertebra in the center of the vertebral bodies. We run the segmentation algorithm for each individual vertebra separately. We refer the interested reader to [37] for more details on the algorithm. For edge detection, we set the parameters to $\alpha = 20$ and $\beta = 0.55$. In our segmentation algorithm, we consider anisotropic edges due to the fine vertebral substructures. The regularization parameters are set to $\lambda = 0.04$ and $\lambda_{shape} = 0.005$ for all experiments. The weighting map f emphasizes bone-like structures as foreground regions. The weighting map f_{shape} contains the registered vertebral shape model. We achieve the binary segmentations by thresholding the segmentation u with a value of 0.2.

5.4.2 Qualitative Results

A qualitative result of a correctly segmented fifth lumbar vertebra is illustrated in Fig. 5.24. Figure 5.25 depicts a correctly segmented third thoracic vertebrae of the same subject. Comparing both figures, we see a significant difference in shape between the vertebrae which emphasizes the use of different vertebral shape models. In contrast to that, Fig. 5.26 illustrates an example for a sixth thoracic vertebra where the segmentation is influenced by connected ribs. Small intervertebral discs are segmented wrongly as depicted in Fig. 5.27 showing the first thoracic vertebra. This problem commonly appears in upper thoracic vertebrae. Our presented framework can also be used for fast correction of wrong segmentations. In this case, the wrong segmentation is used as a weighting map in our framework where. Incorporating foreground and background scribbles allows for a correction within seconds. An example for vertebra correction of the sixth thoracic vertebra is shown in Fig. 5.28.

5.4.3 Quantitative Results

The performance of our algorithm was evaluated by a leave-one-out cross validation on the training data, i.e., we report average performance over ten experiments. A second evaluation was performed online on additional five CT data sets of healthy subjects with unknown reference segmentations. The quantitative evaluation was performed in terms of the DSC and MAD.

We achieve an average DSC of $93\% \pm 4\%$ and MAD of 0.47 ± 0.54 mm over all vertebrae from the leave-one-out experiment. Our algorithm performs well on lumbar vertebrae ($96\% \pm 2\%$) and lower thoracic vertebrae T7-T12 ($95\% \pm 2\%$). The DSC for thoracic vertebrae T1-T6 is $89\% \pm 5\%$, which can be explained by the influence of ribs and small intervertebral discs that are connected to the vertebrae. All estimated cross-validation scores for DSC are depicted in Fig. 5.29. The results for the MAD illustrated in Fig. 5.30 show the same pattern as the DSC. For upper thoracic vertebrae, we achieve an MAD of 0.74 ± 0.75 mm. Lower thoracic (0.30 ± 0.25 mm) and lumbar vertebrae (0.35 ± 0.36 mm) have a comparable MAD. Large standard deviations indicate the large variation in MAD due to the limited number of examples and outliers.

The online evaluation of the test set shows the same pattern as for the cross-validation as depicted in Fig. 5.31. The algorithm performs equally well on lumbar vertebrae ($97\% \pm 1\%$) and lower thoracic vertebrae T7-T12 ($96\% \pm 1\%$), the standard deviation is low. We achieve a DSC of $91\% \pm 4\%$ for upper thoracic vertebrae. Results for the MAD were not provided.

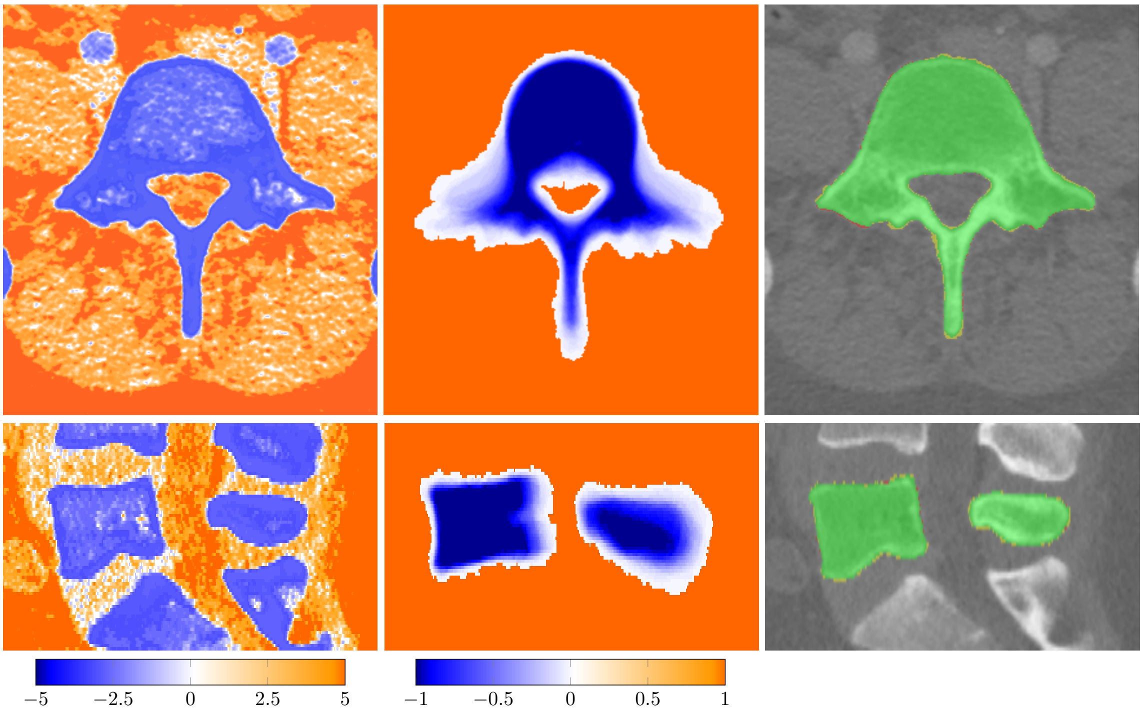


Figure 5.24: The top row shows mid-sagittal cross sections and the bottom row axial cross sections of a fifth lumbar vertebra. Blue regions in the weighting map (left image) are classified as bone. The registered vertebral mean shape (middle image) and the bone weighting map are used as main information for the segmentation algorithm. The final segmentation is depicted in the right images where green regions are correctly segmented. Red regions are wrongly segmented by our algorithm and yellow regions are missing in our segmentation. The DSC for this example is 97%, the MAD equals 0.12 mm.

The evaluation for the individual cases is illustrated in Fig. 5.32 and Fig. 5.33. For all cases except case 6, the average DSC is between $91 - 95\%$ and the average MAD is $0.22 - 0.83$ mm. Case 6 (DSC$= 86\% \pm 9\%$, MAD$=1.4 \pm 1.02$mm) contains many regions corresponding to soft tissue inside the vertebrae, especially in the vertebral bodies. Thus, the mean shape cannot be registered correctly. The final segmentations are not correct either, as they are attracted not only to the bone prior, but also to the mean shape. Furthermore, the registration fails for T12 and L5, hence, they are not included in the calculations of evaluation metrics.

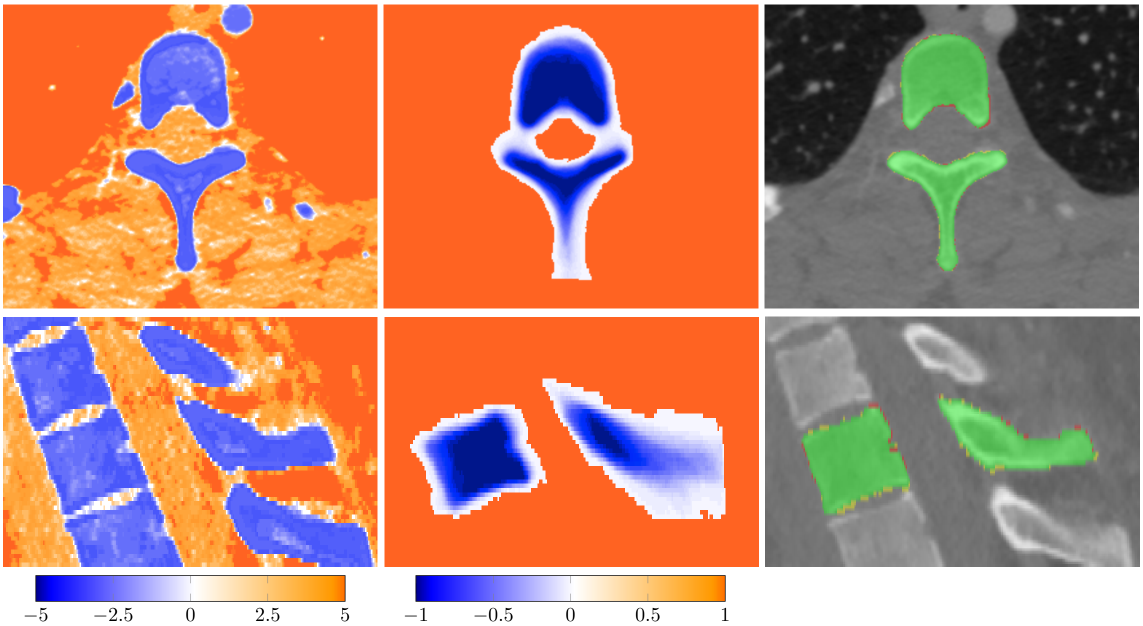


Figure 5.25: The top row shows mid-sagittal cross sections and the bottom row axial cross sections of the thoracic vertebra T3. Blue regions in the weighting map (left image) are classified as bone. The registered mean shape (middle image) and the bone prior are used as main information for the segmentation algorithm. The final segmentation is depicted in the right images where green regions are correctly segmented. Red regions are wrongly segmented by our algorithm and yellow regions are missing in our segmentation. The DSC for this example is 95%, the MAD equals 0.17 mm.

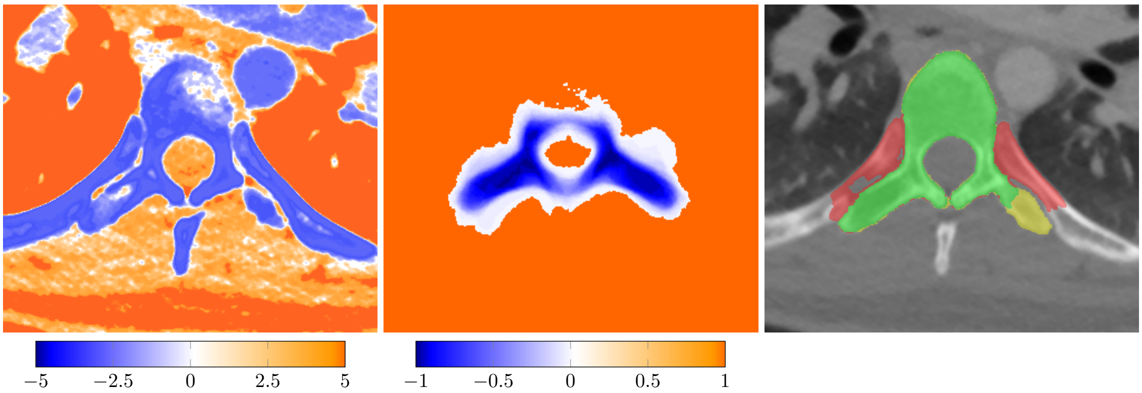


Figure 5.26: This example illustrates axial cross sections of T6 where we achieve a DSC of 84% and an MAD of 1.16 mm. The bone weighting map shown in the left image classifies ribs as bone tissue. Therefore, the registered mean shape (middle image) is wrongly aligned, as it is attracted by the ribs. The final segmentation (right image) contains wrongly segmented ribs illustrated in red, while the yellow parts of the vertebra are missing. The green area depicts the correctly segmented region.

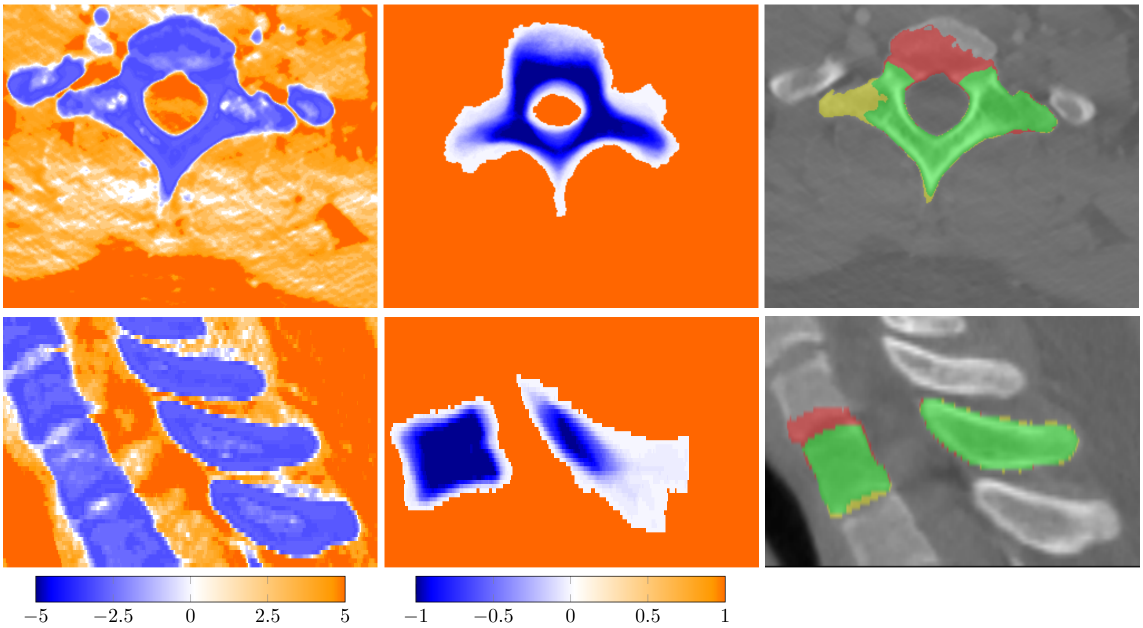
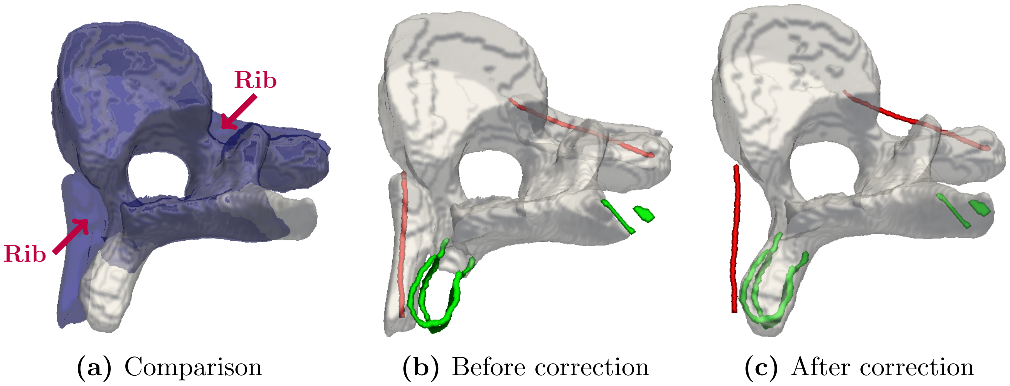


Figure 5.27: This example illustrates axial cross sections of T1 where we achieve a DSC of 84% and an MAD of 0.94 mm. The bone weighting map shown in the left image classifies intervertebral discs as bone tissue. Therefore, the registered mean shape (middle image) is wrongly aligned. The final segmentation (right image) contains wrongly segmented intervertebral discs illustrated in red, while the yellow parts of the vertebra are missing. The green area depicts the correctly segmented region.



(a) Comparison **(b)** Before correction **(c)** After correction

Figure 5.28: This example illustrates manual vertebra correction for the sixth thoracic vertebra. Image (a) compares the vertebrae before and after correction. Unstructered blue areas show the over-segmented ribs. Setting one background scribble (red) in each rib and a few foreground scribbles (green) in the left transverse process and spinous process as depicted in (b) results in a corrected vertebra shown in (c).

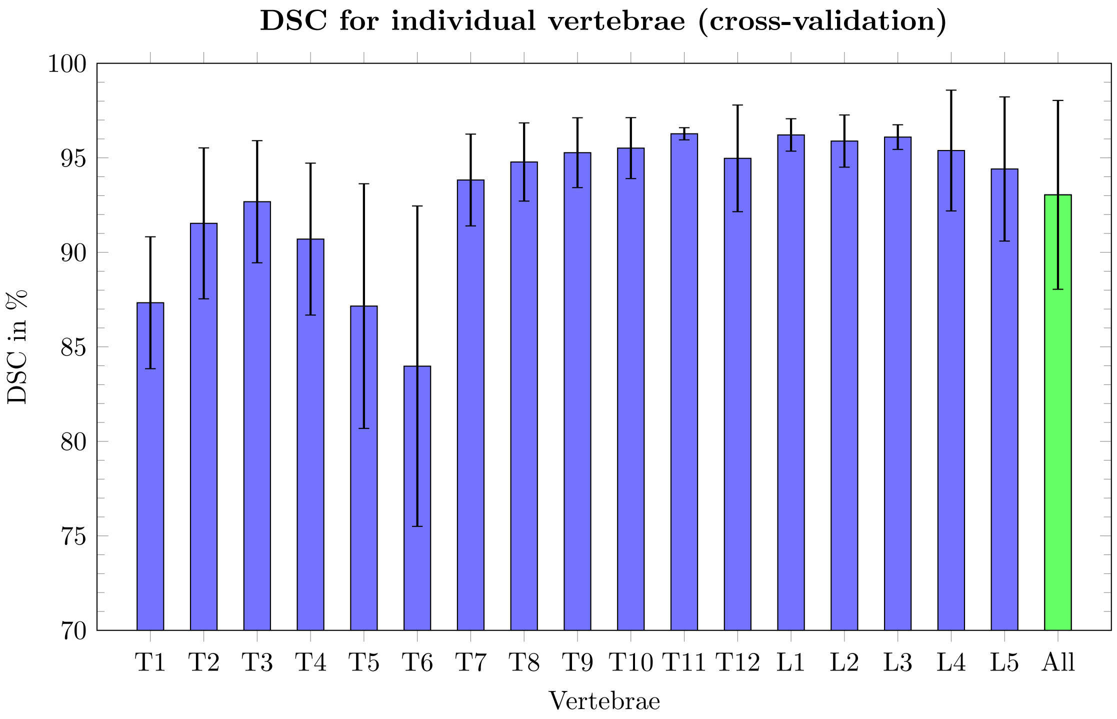


Figure 5.29: DSC for the cross-validated training set.

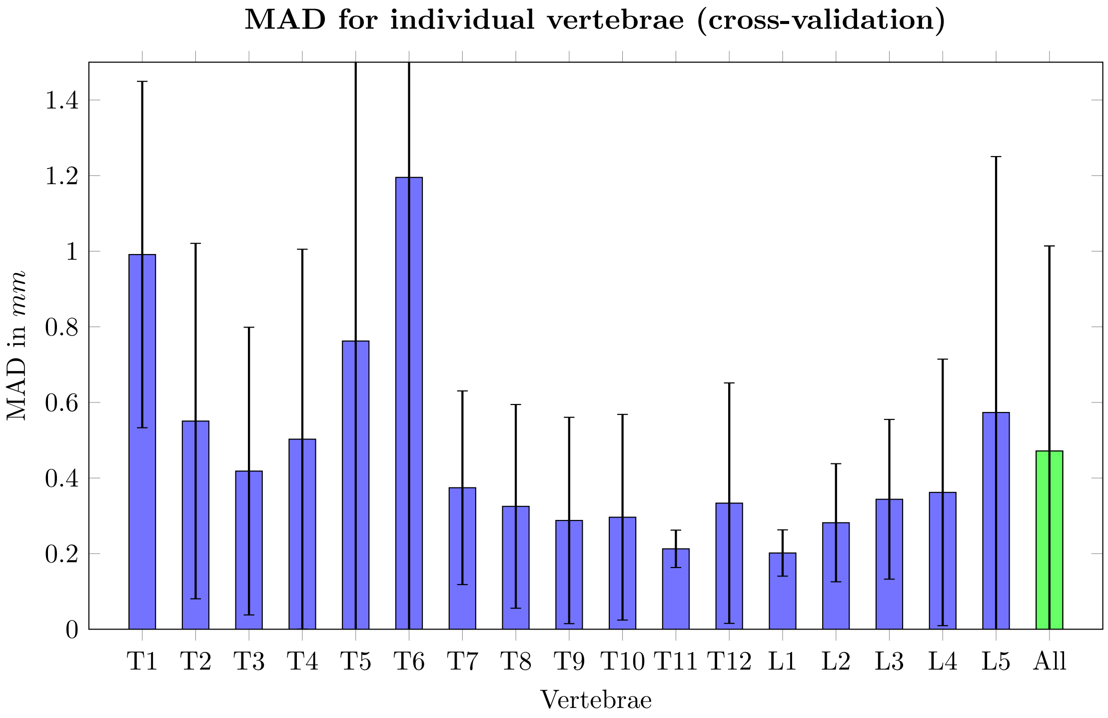


Figure 5.30: MAD for the cross-validated training set.

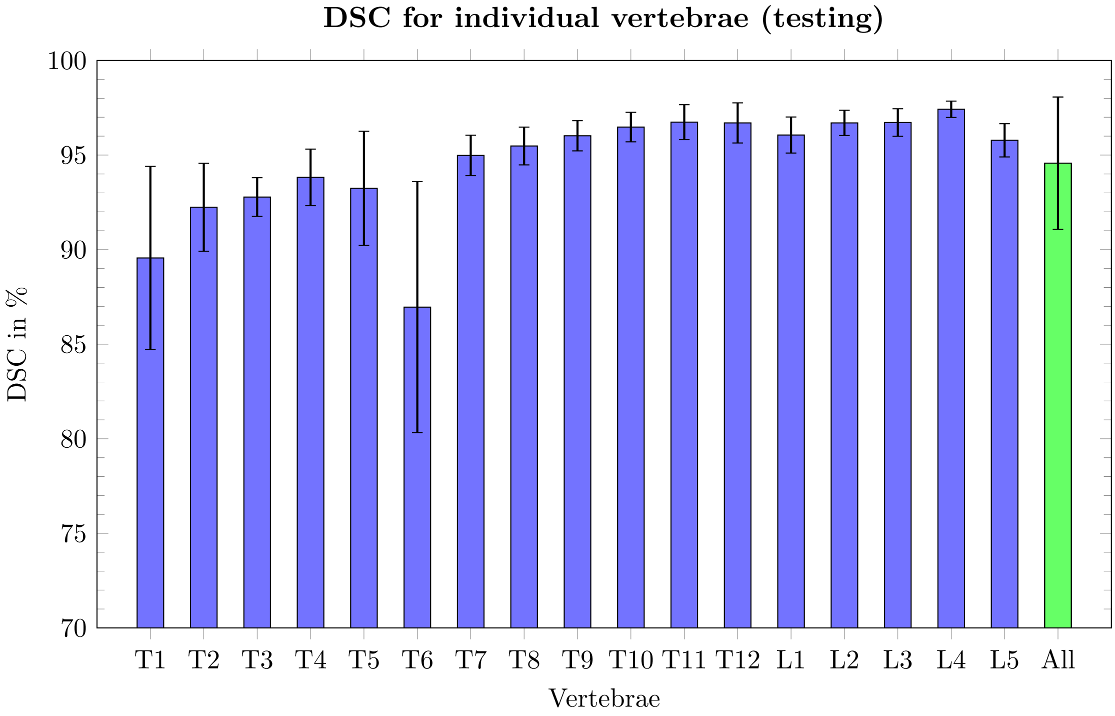


Figure 5.31: DSC for the test set.

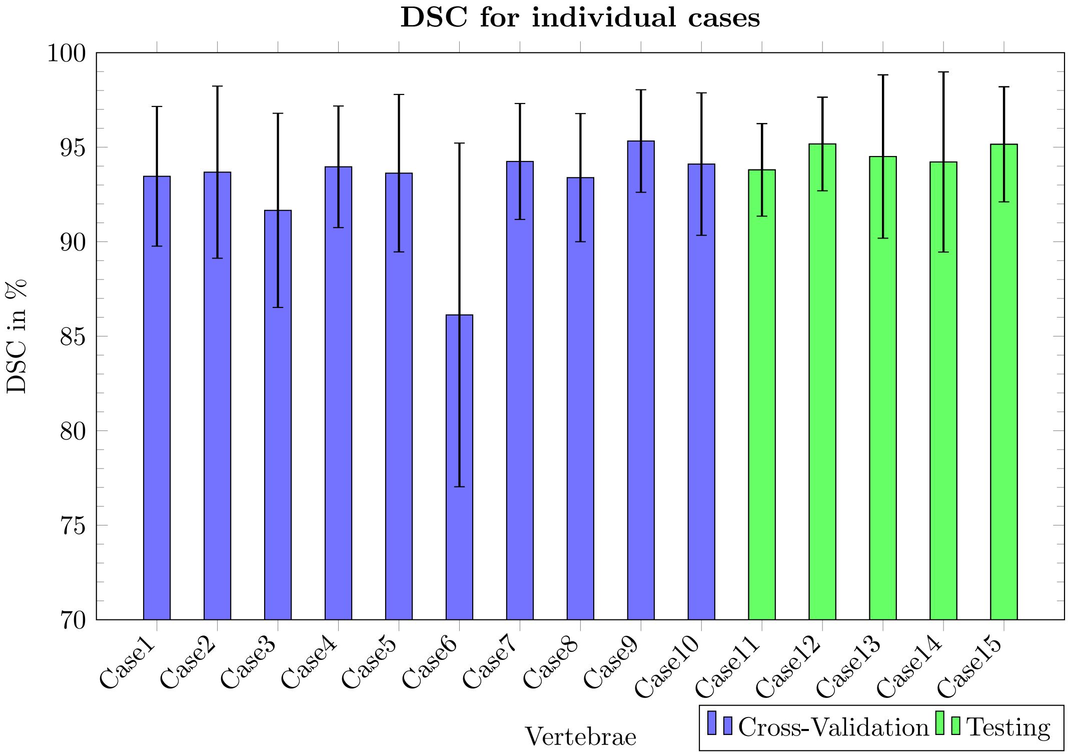


Figure 5.32: DSC for individual test cases.

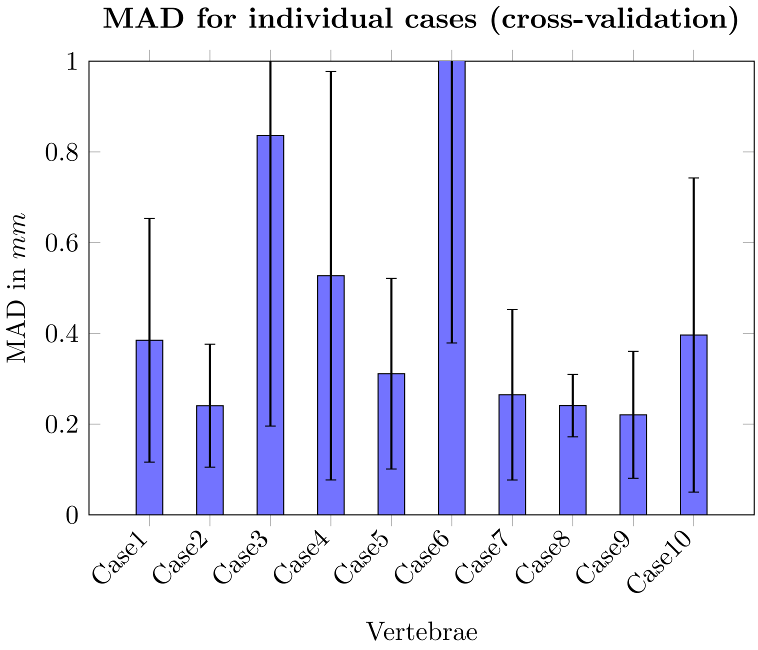


Figure 5.33: MAD for individual test cases.

5.5 Discussion

In this section, we evaluated the impact of global and model specific shape constraints. Prior knowledge in terms of shape constraints are essential to make restrictions on the solution leading to more accurate and robust results. We showed that we can minimize user interaction using shape constraints.

Global Shape Constraints

Global shape constraints need only a single ellipsoid or point initializations as input. All constraints can be extracted from the user-specified ellipsoid. We can also add additional local user constraints in terms of foreground or background scribbles. A weighting map, which is a central element of our segmentation framework, can also be obtained from the ellipsoid information. There exist many ways to obtain a weighting map. Any probability map or classification output can be used. More sophistic weighting maps can further improve segmentation accuracy.

We used the star shape prior [35, 67] and image moment constraints [44, 45]. We studied the impact of image moment constraints in 2D. We saw that using only area or CoG constraint is not enough for a correct segmentation. The success of using these constraints highly depends on the weighting map. We observed that the volume is distributed all over the image if no further constraints are made. Furthermore, objects do not necessarily have to be connected.

The star shape prior is very powerful. It has the nice property to segment one-connected (star) convex objects. To segment more complicated shapes, an extension to multiple star centers is used. Sometimes it is sufficient to use the star prior only. We applied this successfully to 2D images. In many cases, the star prior leads to over-segmentation because it connects to similar structures. We achieve promising results in 2D and 3D when combining the star prior with moment constraints. Especially in 3D, it turned out that combining star prior with volume and CoG constraints is very useful. Experiments showed also that combining star prior and covariance constraint is often too restrictive.

Global shape constraints could also be useful in clinical practice. We studied the application to sinus floor augmentation segmentation. Nowadays, segmentation is still performed manually in this very demanding area. Using manual, local user scribbles combined with GAC segmentation decreases the user interaction time from an hour to a few minutes which is already a high improvement. Using global shape constraints, we could further decrease the amount of user interaction while still achieving promising results. We offer also the possibility to define a cutting plane preventing the segmentation to leak out in lower axial slices, hence, this increases segmentation accuracy. A combination of star prior, CoG and volume constraint turned out to be useful for sinus floor augmentation segmentation. In general, sinus floor augmentation segmentation is a very ambiguous task. It is difficult to assess which tissue is really grafting material and not bone. To study and evaluate global shape constraints more extensively, better experts' reference segmentations have to be provided. Therefore, we suggest to use GAC based image segmentation with local user scribbles in clinical practice.

The global shape constraints are all formulated in terms of convex sets. We showed two algorithms to incorporate these constraints. The PD-Dykstra algorithm includes an additional optimization algorithm for projection onto convex sets and in our case needs to compute the pseudo-inverse for CoG and covariance constraint, which is computationally demanding. Furthermore, depending on the complexity of convex sets, it is difficult to find the intersection of these sets. We showed how to incorporate the constraints efficiently and exactly using the PD algorithm. While Klodt et al. [44] used a numerical approximation based on a lagged diffusivity approach combined with Dykstra's algorithm to project onto the intersection of convex set which is not clear that it converges to the optimal solution our algorithm allows for an exact solution of shape constraints. Preconditioning is essential for proper convergence. We showed that a good convergence is achieved for the preconditioning parameters $\alpha = 2$ and $\xi = 1$.

We incorporated the moment constraints as equality constraints, thus, the constraints are very restrictive because they have to be fulfilled exactly. An extension of our algorithm would be to study the impact of moment constraints in a weak sense, i.e., allow for a slight deviation. We suggest to define an open ball constraint with radius ϵ for moment constraints.

Model Specific Shape Constraints

The second type of shape constraints that we studied are model specific shape constraints on the example of vertebrae segmentation. Our proposed method for vertebrae segmentation based on a variational framework has been successfully applied to ten volumetric CT data sets provided for the CSI spine and vertebrae segmentation challenge. We incorporate a successfully registered mean shape model in our segmentation framework guiding the segmentation algorithm. A common problem in vertebrae segmentation is that edges of the vertebrae are not clearly defined and that connected ribs may get over-segmented. Furthermore, trabecular bone intensities sometimes resemble soft tissue. The overall result of $93\% \pm 4\%$ in terms of the DSC is similar to other published methods.

The result in lumbar region ($96\% \pm 2\%$) is better than 95% presented by Kadoury et al. [40] and the result of Ibragimov et al. [39] evaluated only on the lumbar vertebrae ($93\% \pm 2\%$). In the lower thoracic part of the spine, our result of $95\% \pm 2\%$ exceeds the overall result for thoracic vertebrae (i.e. 93%) presented by Kadoury et al. [40]. However, these methods were evaluated on different data sets, so results are not fully comparable. A direct comparison to four other methods that were evaluated on the same dataset was done at the MICCAI vertebrae segmentation challenge where our method won the second prize (see Appendix C).

The lower DSC values for thoracic vertebrae T1-T6 can be explained by the influence of ribs and small intervertebral discs that mislead the segmentation. Again, the weighting map is an essential part of the algorithm, hence, it also has a high impact. Improving the weighting map using more sophisticated methods, e.g. random forest classification [59], could further improve the segmentation results.

6

Conclusion and Outlook

Contents

6.1 Conclusion

In this master's thesis, we studied the theory and impact of different kinds of shape constraints and their applicability to 2D and 3D image segmentation problems. Prior knowledge in terms of shape constraints is necessary to achieve robust and accurate segmentation results. Furthermore, they allow to decrease the amount of user interaction in interactive segmentation methods.

We incorporated the shape constraints in a powerful Geodesic Active Contours (GAC) based variational segmentation framework formulated as a convex energy minimization problem. Using convex functionals and convex sets yields globally optimal solutions independent of initialization. Furthermore, we used an efficient primal-dual algorithm to solve non-smooth problems and exploited its high parallelization potential on the Graphics Processing Unit (GPU). Compared to Dykstra's projection algorithm, we showed how we can incorporate the constraints directly in the primal-dual framework resulting in an exact solution of constraints. Furthermore, we overcome computationally expensive calculations such as matrix inversions.

Our segmentation framework is very versatile. It can be easily adapted to other constraints. On the one hand, we showed the application-dependent incorporation of specific shape models on the example of vertebra segmentation. The algorithm only needs an initialization in the vertebral bodies. The quantitative segmentation results are promising compared to literature. On the other hand, using global shape constraints is not application specific. Global shape constraints require a single ellipsoid input or manual point initializations. Both moment constraints and star prior constraint have disadvantages when they are applied solely. Combining them results in a powerful tool. We applied a combination of star prior with area and Center of Gravity (CoG) constraints successfully to a number of 2D and 3D examples, including brain tumor and maxillary sinus floor grafting material segmentation.

6.2 Outlook

A possible future work would be to improve the segmentation GUI in 3D. By now, the user cannot observe the impact of shape constraints in every iteration. We could enhance the GUI such that the user gets immediate feedback which allows for fast additional user interaction.

We tested the model specific shape prior on vertebra segmentation, but it can be also useful for a number of other task. Furthermore, we plan to test the algorithm on pathological data if training data is available.

In this work, we used multiple star centers in combination with moment constraints. It would be interesting to extend this to geodesic stars as proposed by [35] to further reduce user interaction and describe more complicated shapes easier. In medical image segmentation, the application of multiple star centers in combination with moment constraints could be studied in 3D for segmentation of complex organs showing a large degree of variations, e.g. the liver. Before global shape constraints are used in clinical practice, a detailed evaluation has to be performed based on accurate experts' segmentations.

Besides the shape constraints, weighting maps are an important part of our segmentation framework. Improving the weighting maps using sophisticated machine learning algorithms may also lead to better segmentation results.

A

List of Acronyms

ASM	Active Shape Model
CoG	Center of Gravity
CT	Computed Tomography
DSC	Dice Similarity Coefficient
FoV	Field of View
GAC	Geodesic Active Contours
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HD	Hausdorff Distance
HU	Hounsfield Units
ISTA	Iterative Shrinkage and Thresholding Algorithm
l.s.c.	lower-semicontinuous
MAD	Mean Absolute Surface Distance
MAP	Maximum A Posteriori
MRI	Magnetic Resonance Imaging
PCA	Principal Component Analysis
PET	Positron Emission Tomography
RF	Radio Frequency
SNR	Signal-to-Noise Ratio
SSM	Statistical Shape Model
TV	Total Variation
US	Ultrasound

B

Upper Bounds for the Norm of Derivative Operators

In the following, we show how we can derive an upper bound for the linear gradient operator $\nabla : X \mapsto Y$ in 2D and 3D. The divergence operator $\operatorname{div} : Y \mapsto X$ is related to the gradient operator by $\langle \nabla u, p \rangle = \langle u, -\operatorname{div} p \rangle$, the operator norm is the same. We estimate a constant L that bounds both operators from above with the same value. In MATLAB, L can be estimated using the command `normest`.

$$\|\nabla u\|_Y \leq L\|u\|_X, \quad u \in X \tag{B.1}$$

$$\|\operatorname{div} p\|_X \leq L\|p\|_Y, \quad p \in Y \tag{B.2}$$

For the derivation we operate in the discrete domain as outlined in Section 2.3. We recall the norms defined for a scalar field u and a vector field p:

- Norm of a scalar field u: $\|u\|_X = \sqrt{\sum_{\forall \mathbf{i}} u_{\mathbf{i}}^2}$
- Norm of a vector field p: $\|p\|_Y = \sqrt{\sum_{\forall \mathbf{i}} \left((p_{\mathbf{i}}^x)^2 + (p_{\mathbf{i}}^y)^2 + (p_{\mathbf{i}}^z)^2\right)}$

Additionally, we use following property for the derivation:

$$\|a + b\|^2 = \|a\|^2 + 2\langle a, b \rangle + \|b\|^2 \tag{B.3}$$

Using Young's inequality, we get:

$$\|a\|^2 + 2\langle a, b \rangle + \|b\|^2 \leq \|a\|^2 + \|b\|^2 + \frac{\|a\|^2}{\epsilon} + \epsilon\|b\|^2, \ \epsilon > 0 \tag{B.4}$$

Setting $\epsilon = 1$ yields:

$$\|a + b\|^2 \leq 2\left(\|a\|^2 + \|b\|\right) \tag{B.5}$$

B.1 Derivation in 2D

We derive the norm of the gradient in two dimensions. The gradient operator is bounded from above by a constant L as follows:

$$\|\nabla_{2D}u\|_Y \leq L\|u\|_X$$
$$\|\nabla_{2D}u\|_Y^2 \leq L^2\|u\|_X^2$$

Now, we discretize the left-hand side by using forward differences δ^+:

$$\|\nabla_{2D}u\|_Y^2 = \sum_{\substack{1\leq i\leq M\\1\leq j\leq N}} (\nabla u_{i,j})^2 = \sum_{\substack{1\leq i\leq M\\1\leq j\leq N}} \left(\frac{u_{i+1,j}-u_{i,j}}{h_x}\right)^2 + \left(\frac{u_{i,j+1}-u_{i,j}}{h_y}\right)^2$$

The Neumann boundary conditions state that $u_{M+1,j}-u_{M,j}=0$ and $u_{i,N+1}-u_{i,N}=0$, thus the limits for the sum are decreased by one.

$$\sum_{\substack{1\leq i\leq M-1\\1\leq j\leq N-1}} \left(\frac{u_{i+1,j}-u_{i,j}}{h_x}\right)^2 + \left(\frac{u_{i,j+1}-u_{i,j}}{h_y}\right)^2 \leq$$
$$\sum_{\substack{1\leq i\leq M-1\\1\leq j\leq N-1}} \frac{2}{h_x^2}\left(u_{i+1,j}^2+u_{i,j}^2\right) + \frac{2}{h_y^2}\left(u_{i,j+1}^2+u_{i,j}^2\right) \leq$$
$$\sum_{\substack{1\leq i\leq M\\1\leq j\leq N}} \frac{4}{h_x^2}\left(u_{i,j}^2\right) + \frac{4}{h_y^2}\left(u_{i,j}^2\right) = 4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\sum_{\substack{1\leq i\leq M\\1\leq j\leq N}} u_{i,j}^2 =$$
$$4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\|u\|_X^2 = L^2\|u\|_X^2$$

We estimate the upper bound L as:

$$L = \sqrt{4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)} \tag{B.6}$$

Depending on the image spacings h_x, h_y we can specify a constant L for the case of anisotropic, isotropic and unit voxels:

- Anisotropic Voxels $h_x \neq h_y$: $L = \sqrt{4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)}$
- Isotropic Voxels $h_x = h_y = h$: $L = \sqrt{\frac{8}{h^2}}$
- Unit Voxels $h_x = h_y = 1$: $L = \sqrt{8}$

B.2 Derivation in 3D

We derive the norm of the gradient operator in three dimensions in the same manner as for two dimensions. First, we discretize the Total Variation (TV) norm by using forward differences δ^+:

$$\|\nabla_{3D} u\|^2 = \sum_{\substack{1\le i\le M\\1\le j\le N\\1\le k\le D}} (\nabla u_{i,j,k})^2 =$$

$$\sum_{\substack{1\le i\le M\\1\le j\le N\\1\le k\le D}} \left(\frac{u_{i+1,j,k}-u_{i,j,k}}{h_x}\right)^2 + \left(\frac{u_{i,j+1,k}-u_{i,j,k}}{h_y}\right)^2 + \left(\frac{u_{i,j,k+1}-u_{i,j,k}}{h_z}\right)^2$$

The Neumann boundary conditions state that $u_{M+1,j,k}-u_{M,j,k}=0$, $u_{i,N+1,k}-u_{i,N,k}=0$ and $u_{i,j,D+1}-u_{i,j,D}=0$, thus the limits for the sum are decreased by one.

$$\sum_{\substack{1\le i\le M-1\\1\le j\le N-1\\1\le k\le D-1}} \left(\frac{u_{i+1,j,k}-u_{i,j,k}}{h_x}\right)^2 + \left(\frac{u_{i,j+1,k}-u_{i,j,k}}{h_y}\right)^2 + \left(\frac{u_{i,j,k+1}-u_{i,j,k}}{h_z}\right)^2 \le$$

$$\sum_{\substack{1\le i\le M-1\\1\le j\le N-1\\1\le k\le D-1}} \frac{2}{h_x^2}\left(u_{i+1,j,k}^2+u_{i,j,k}^2\right) + \frac{2}{h_y^2}\left(u_{i,j+1,k}^2+u_{i,j,k}^2\right) + \frac{2}{h_z^2}\left(u_{i,j,k+1}^2+u_{i,j,k}^2\right) \le$$

$$\sum_{\substack{1\le i\le M\\1\le j\le N\\1\le k\le D}} \frac{4}{h_x^2}\left(u_{i,j,k}^2\right) + \frac{4}{h_y^2}\left(u_{i,j,k}^2\right) + \frac{4}{h_z^2}\left(u_{i,j,k}^2\right) = 4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}+\frac{1}{h_z^2}\right) \sum_{\substack{1\le i\le M\\1\le j\le N\\1\le k\le D}} u_{i,j,k}^2 =$$

$$4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)\|u\|_X^2 = L^2\|u\|_X^2$$

We estimate the upper bound L as:

$$L = \sqrt{4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)} \tag{B.7}$$

Depending on the image spacings h_x, h_y, h_z we can specify a constant L for the case of anisotropic, isotropic and unit voxels:

- Anisotropic Voxels $h_x \neq h_y \neq h_z$: $L = \sqrt{4\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)}$
- Isotropic Voxels $h_x = h_y = h_z = h$: $L = \sqrt{\frac{12}{h^2}}$
- Unit Voxels $h_x = h_y = h_z = 1$: $L = \sqrt{12}$

C

Publication

My master's thesis at the Institute for Computer Graphics and Vision led to the following publication. It is listed along with the respective abstract.

Vertebrae Segmentation in 3D CT Images based on a Variational Framework

Kerstin Hammernik, Thomas Ebner, Darko Stern, Martin Urschler and Thomas Pock

In: *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging* (To be published in 2015)

This work was accepted for oral presentation on the second MICCAI workshop for "Computational Methods and Clinical Applications for Spine Imaging" held at Harvard Medical School, Boston, MA, USA on September 14^{th}, 2014.

With this work, we took part on the 2014 Vertebra Segmentation Challenge. We won the second prize and were given an honorable mention award.

Abstract: Automatic segmentation of 3D vertebrae is a challenging task in medical imaging. In this paper, we introduce a total variation (TV) based framework that incorporates an a priori model, i.e., a vertebral mean shape, image intensity and edge information. The algorithm was evaluated using leave-one-out cross validation on a data set containing ten computed tomography scans and ground truth segmentations provided for the CSI MICCAI 2014 spine and vertebrae segmentation challenge. We achieve promising results in terms of the Dice Similarity Coefficient (DSC) of 0.93 ± 0.04 averaged over the whole data set.

Bibliography

[1] Adams, R. and Bischof, L. (1994). Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647. (page 4)

[2] Beck, A. (2014). *Introduction to Nonlinear Optimization*. Society for Industrial and Applied Mathematics. (page 23)

[3] Beck, A. and Teboulle, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202. (page 32)

[4] Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific. (page 23, 29)

[5] Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. In *European Conference on Computer Vision (ECCV)*. (page 4)

[6] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA. (page 23)

[7] Boykov, Y. and Jolly, M.-P. (2000). Interactive Organ Segmentation using Graph Cuts. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 1935 of *Lecture Notes in Computer Science*, pages 276–286. Springer, Springer Verlag. (page 4)

[8] Boykov, Y. and Jolly, M.-P. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *International Conference on Computer Vision (ICCV)*, pages 105–112. (page 4)

[9] Boykov, Y. and Kolmogorov, V. (2003). Computing Geodesics and Minimal Surfaces via Graph Cuts. In *IEEE International Conference on Computer Vision (ICCV)*. (page 19)

[10] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239. (page 3)

[11] Boyle, J. P. and Dykstra, R. L. (1986). A Method for Finding Projections onto the Intersection of Convex Sets in Hilbert Spaces. In *Advances in Order Restricted Statistical Inference*, volume 37 of *Lecture Notes in Statistics*, pages 28–47. Springer New York. (page 29)

[12] Bredies, K., Kunisch, K., and Pock, T. (2010). Total Generalized Variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526. (page 15)

[13] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.-P., and Oster, S. (2007). Fast Global Minimization of the Active Contour/Snake Model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167. (page 17, 18)

[14] Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79. (page 3, 17)

[15] Chambolle, A. (2004). An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97. (page 21)

[16] Chambolle, A. and Lions, P.-L. (1997). Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188. (page 15)

[17] Chambolle, A. and Pock, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145. (page 33, 34)

[18] Chan, T. F. and Esedoglu, S. (2005). Aspects of Total Variation Regularized L1 Function Approximation. *SIAM Journal on Applied Mathematics*, 66(5):1817–1837. (page 15)

[19] Chan, T. F. and Vese, L. A. (2001). Active Contours without Edges. *IEEE Transactions on Image Processing*, 10(2):266–277. (page 3)

[20] Comaniciu, D. and Meer, P. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619. (page 3)

[21] Combettes, P. L. and Wajs, V. R. (2005). Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200. (page 32)

[22] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active Appearance Models. In Burkhardt, H. and Neumann, B., editors, *European Conference on Computer Vision (ECCV)*, LNCS, pages 483–498. Springer. (page 5)

[23] Cootes, T. F., Hill, A., Taylor, C. J., and Haslam, J. (1994). The Use of Active Shape Models For Locating Structures in Medical Images. *Image and Vision Computing*, 12(6):355–366. (page 4, 5, 49)

[24] Cremers, D., Kohlberger, T., and Schnörr, C. (2002). Nonlinear shape statistics in Mumford–Shah based segmentation. In *European Conference on Computer Vision (ECCV)*. (page 5)

[25] Cremers, D., Osher, S. J., and Soatto, S. (2006). Kernel Density Estimation and Intrinsic Alignment for Shape Priors in Level Set Segmentation. *International Journal of Computer Vision*, 69(3):335–351. (page 5)

[26] Cremers, D., Schmidt, F., and Barthel, F. (2008). Shape Priors in Variational Image Segmentation: Convexity, Lipschitz Continuity and Globally Optimal Solutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 5)

[27] Daubechies, I., Defrise, M., and De Mol, C. (2004). An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457. (page 32)

[28] Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302. (page 53)

[29] Donoser, M., Bischof, H., and Wiltsche, M. (2006). Color Blob Segmentation by MSER Analysis. In *IEEE International Conference on Image Processing (ICIP)*, pages 757–760. (page 20, 59)

[30] Federer, H. (1969). *Geometric measure theory*. Die Grundlehren der mathematischen Wissenschaften. Springer. (page 15)

[31] Gaffke, N. and Mathar, R. (1989). A Cyclic Projection Algorithm Via Duality. *Metrika*, 36(1):29–54. (page 29, 31)

[32] Glocker, B., Zikic, D., Konukoglu, E., Haynor, D. R., and Criminisi, A. (2013). Vertebrae Localization in Pathological Spine CT via Dense Classification from Sparse Annotations. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 262–270. (page 9)

[33] Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 3rd edition. (page 2, 3)

[34] Goodall, C. (1991). Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society B*, 53(2):285–339. (page 4)

[35] Gulshan, V., Rother, C., Criminisi, A., Blake, A., and Zisserman, A. (2010). Geodesic Star Convexity for Interactive Image Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 4, 48, 49, 77, 82)

[36] Hadamard, J. (1902). Sur les problèmes aux Dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13(13):49–52. (page 11)

[37] Hammernik, K., Ebner, T., Stern, D., Urschler, M., and Pock, T. (2015). Vertebrae Segmentation in 3D CT Images based on a Variational Framework. In Yao, J., Glocker, B., Klinder, T., and Li, S., editors, *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, volume 20 of *Lecture Notes in Computational Vision and Biomechanics*. Springer. To be published. (page 50, 70)

[38] Huttenlocher, D. P., Klanderman, G. A., Kl, G. A., and Rucklidge, W. J. (1993). Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863. (page 53)

[39] Ibragimov, B., Likar, B., Pernus, F., and Vrtovec, T. (2014). Shape Representation for Efficient Landmark-Based Segmentation in 3-D. *IEEE Transactions on Medical Imaging*, 33(4):861–874. (page 79)

[40] Kadoury, S., Labelle, H., and Paragios, N. (2011). Automatic inference of articulated spine models in CT images using high-order Markov Random Fields. *Medical Image Analysis*, 15(4):426–437. (page 79)

[41] Kass, M., Witkin, A. P., and Terzopoulos, D. (1988). Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331. (page 3, 16)

[42] Kendall, D. G. (1977). The diffusion of shape. *Advances in Applied Probability*, 9:428–430. (page 4)

[43] Kirmeier, R., Payer, M., Wehrschuetz, M., Jakse, N., Platzer, S., and Lorenzoni, M. (2008). Evaluation of Three-Dimensional Changes after Sinus Floor Augmentation with Different Grafting Materials. *Clinical oral implants research*, 19(4):366–72. (page 7, 66)

[44] Klodt, M. and Cremers, D. (2011). A Convex Framework for Image Segmentation with Moment Constraints. In *IEEE International Conference on Computer Vision (ICCV)*. (page 4, 42, 77, 78)

[45] Klodt, M., Steinbruecker, F., and Cremers, D. (2013). Moment Constraints in Convex Optimization for Segmentation and Tracking. In *Advanced Topics in Computer Vision*. Springer. (page 42, 77)

[46] Leventon, M. E., Grimson, W. E. L., and Faugeras, O. D. (2000). Statistical Shape Influence in Geodesic Active Contours. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 5)

[47] Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M.-A., Arbel, T., Avants, B. B., Ayache, N., Buendia, P., Collins, D. L., Cordier, N., Corso, J. J., Criminisi, A., Das, T., Delingette, H., Demiralp, C., Durst, C. R., Dojat, M., Doyle, S., Festa, J., Forbes, F., Geremia, E., Glocker, B., Golland, P., Guo, X., Hamamci, A., Iftekharuddin, K. M., Jena, R., John, N. M., Konukoglu, E., Lashkari, D., Mariz, J. A., Meier, R., Pereira, S., Precup, D., Price, S. J., Raviv, T. R., Reza, S. M. S., Ryan, M., Sarikaya, D., Schwartz, L., Shin, H.-C., Shotton, J., Silva, C. A., Sousa, N., Subbanna, N. K., Szekely, G., Taylor, T. J., Thomas, O. M., Tustison, N. J., Unal, G., Vasseur, F., Wintermark, M., Ye, D. H., Zhao, L., Zhao, B., Zikic, D., Prastawa, M., Reyes, M., and

Leemput, K. V. (2014). The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*. (page 2, 54, 64)

[48] Mumford, D. and Shah, J. (1989). Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685. (page 3)

[49] Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer. (page 23, 26, 32)

[50] Nikolova, M. (2004). A Variational Approach to Remove Outliers and Impulse Noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120. (page 15, 16)

[51] Pock, T. (2008). *Fast Total Variation for Computer Vision*. PhD thesis, Graz University of Technology, Institute for Computer Graphics and Vision. (page 4, 13)

[52] Pock, T. and Chambolle, A. (2011). Diagonal Preconditioning for First Order Primal-Dual Algorithms in Convex oOptimization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1762–1769. (page 35)

[53] Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2009). An Algorithm for Minimizing the Mumford-shah Functional. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1133–1140. (page 3)

[54] Reinbacher, C., Pock, T., Bauer, C., and Bischof, H. (2010). Variational Segmentation of Elongated Volumetric S tructures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3177–3184. (page 7, 17, 19)

[55] Rockafellar, R. T. (1970). *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press. (page 23)

[56] Rother, C., Kolmogorov, V., and Blake, A. (2004). GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):309–314. (page 4)

[57] Rousson, M. and Paragios, N. (2008). Prior Knowledge, Level Set Representations & Visual Grouping. *International Journal of Computer Vision*, 76(3):231–243. (page 5)

[58] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D*, 60(1-4):259–268. (page 14)

[59] Santner, J., Unger, M., Pock, T., Leistner, C., Saffari, A., and Bischof, H. (2009). Interactive Texture Segmentation using Random Forests and Total Variation. In *Proceedings of the British Machine Vision Conference (BMVC)*. (page 4, 79)

[60] Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition. (page 3)

[61] Shefi, R. and Teboulle, M. (2014). Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, 24(1):269–297. (page 57)

[62] Slabaugh, G. and Unal, G. (2005). Graph Cuts Segmentation using an Elliptical Shape Prior. In *IEEE International Conference on Image Processing (ICIP)*. (page 4)

[63] Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis and Machine Vision*. Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 2nd edition. (page 3)

[64] Tikhonov, A. (1963). Regularization of Incorrectly Posed Problems. *Soviet Mathematics Doklady*, 4:1624–1627. (page 11, 13)

[65] Tikhonov, A. and Arsenin, V. (1977). *Solutions of Ill-Posed Problems*. Winston, Washington D.C. (page 13)

[66] Unger, M., Pock, T., Trobin, W., Cremers, D., and Bischof, H. (2008). TVSeg - Interactive Total Variation Based Image Segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*. (page 4, 7, 19, 20)

[67] Veksler, O. (2008). Star Shape Prior for Graph-Cut Image Segmentation. In *European Conference on Computer Vision (ECCV)*. (page 4, 48, 77)

[68] Vicente, S., Kolmogorov, V., and Rother, C. (2008). Graph Cut Based Image Segmentation with Connectivity Priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 4)

[69] Weickert, J. (1998). *Anisotropic Diffusion In Image Processing*. Teubner, Stuttgart. (page 17)

[70] Werlberger, M., Pock, T., Unger, M., and Bischof, H. (2009). A Variational Model for Interactive Shape Prior Segmentation and Real-Time Tracking. In *Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 5567 of *Lecture Notes in Computer Science*, pages 200–211. Springer Berlin Heidelberg. (page 4, 7, 19)

[71] Xu, C. and Prince, J. L. (1997). Gradient Vector Flow: A New External Force for Snakes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 3, 16)

[72] Yao, J., Burns, J. E., Getty, S., Stieger, J., and Summers, R. M. (2014). Automated Extraction of Anatomic Landmarks on Vertebrae based on Anatomic Knowledge and Geometrical Constraints. In *IEEE International Symposium on Biomedical Imaging (ISBI),*, pages 397–400. (page 9)

[73] Yao, J., Burns, J. E., Munoz, H., and Summers, R. M. (2012). Detection of Vertebral Body Fractures Based on Cortical Shell Unwrapping. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. (page 9, 70)