The Probabilistic Active Shape Model: From Model Construction to Flexible Medical Image Segmentation

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Matthias Kirschner, M. Sc.

geboren in Warburg, Deutschland

Referenten der Arbeit:	Prof. Dr. techn. Dieter W. Fellner Technische Universität Darmstadt Prof. Dr. sc. hum. Hans-Peter Meinzer Deutsches Krebsforschungszentrum
Tag der Einreichung:	13.05.2013
Tag der mündlichen Prüfung:	04.07.2013

Erscheinungsjahr 2013

Darmstädter Dissertation
D 17

Erklärung zur Dissertation

Hiermit versichere ich die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 15.07.2013 Matthias Kirschner

Abstract

Automatic processing of three-dimensional image data acquired with computed tomography or magnetic resonance imaging plays an increasingly important role in medicine. For example, the automatic segmentation of anatomical structures in tomographic images allows to generate three-dimensional visualizations of a patient's anatomy and thereby supports surgeons during planning of various kinds of surgeries.

Because organs in medical images often exhibit a low contrast to adjacent structures, and because the image quality may be hampered by noise or other image acquisition artifacts, the development of segmentation algorithms that are both robust and accurate is very challenging. In order to increase the robustness, the use of model-based algorithms is mandatory, as for example algorithms that incorporate prior knowledge about an organ's shape into the segmentation process. Recent research has proven that Statistical Shape Models are especially appropriate for robust medical image segmentation. In these models, the typical shape of an organ is learned from a set of training examples. However, Statistical Shape Models have two major disadvantages: The construction of the models is relatively difficult, and the models are often used too restrictively, such that the resulting segmentation does not delineate the organ exactly.

This thesis addresses both problems: The first part of the thesis introduces new methods for establishing correspondence between training shapes, which is a necessary prerequisite for shape model learning. The developed methods include consistent parameterization algorithms for organs with spherical and genus 1 topology, as well as a nonrigid mesh registration algorithm for shapes with arbitrary topology. The second part of the thesis presents a new shape model-based segmentation algorithm that allows for an accurate delineation of organs. In contrast to existing approaches, it is possible to integrate not only linear shape models into the algorithm, but also nonlinear shape models, which allow for a more specific description of an organ's shape variation.

The proposed segmentation algorithm is evaluated in three applications to medical image data: Liver and vertebra segmentation in contrast-enhanced computed tomography scans, and prostate segmentation in magnetic resonance images.

Zusammenfassung

Die computergestützte Verarbeitung dreidimensionaler Bilddaten aus der Computer- oder Magnetresonanztomographie spielt eine immer wichtigere Rolle in der Medizin. So kann die automatische Segmentierung anatomischer Strukturen in den Bilddaten dabei helfen, die Anatomie eines Patienten dreidimensional zu visualisieren und somit Chirurgen bei der Planung verschiedenster Eingriffe zu unterstützen.

Da Organe in medizinischen Bilddaten häufig einen geringen Kontrast zueinander aufweisen und die Bildqualität zudem durch Rauschen oder diverse andere Aufnahmeartefakte beeinträchtigt sein kann, ist die Entwicklung von Segmentierungsverfahren, die sowohl zuverlässig als auch genau arbeiten, äußerst schwierig. Um eine hohe Robustheit zu erreichen, ist eine modellbasierte Segmentierung unabdingbar, bei der etwa Vorwissen über die Form eines Organs in den Segmentierungsprozess eingebunden wird. Die bisherige Forschung hat gezeigt, dass sich insbesondere statistische Formmodelle für die robuste Segmentierung medizinischer Bilder eignen. Bei diesen Modellen wird die typische Form eines Organs aus einer Menge von Trainingsdaten gelernt. Allerdings haben statistische Formmodelle zwei wesentliche Nachteile: Zum einen ist der Aufbau der Modelle relativ aufwändig, und zum anderen werden die Modelle häufig zu restriktiv eingesetzt, so dass die resultierende Segmentierung das Organs nicht genau umrandet.

Diese Arbeit adressiert beide Nachteile: Im ersten Teil der Arbeit werden neue Methoden vorgestellt, um Korrespondenzen zwischen Trainingsformen herzustellen, was für den Aufbau von Formmodellen zwingend erforderlich sind. Die entwickelten Methoden umfassen konsistente Parametrisierungsverfahren für Organe mit sphärischer und toroidaler Topologie, sowie ein nicht-rigides Registrierungsverfahren für Organe mit beliebiger Topologie. Der zweite Teil der Arbeit stellt einen neuen Segmentierungsalgorithmus vor, der eine präzise formmodellbasierte Segmentierung ermöglicht. Im Gegensatz zu bestehenden Algorithmen ist das Verfahren in der Lage, neben den sonst üblichen linearen Formmodellen auch nichtlineare Formmodelle zu integrieren, so dass die Formvariation von Organen spezifischer modelliert werden kann.

Das vorgeschlagene Segmentierungsverfahren wird an drei klinischen Problemstellungen evaluiert: Leber- und Wirbelknochensegmentierung in kontrastverstärkten Aufnahmen aus der Computertomographie, sowie Prostatasegmentierung in Aufnahmen aus der Magnetresonanztomographie.

Contents

1. Introduction **1**
1.1. Contributions 5
1.2. Structure of this thesis and notation 6

2. State of the art **9**
2.1. Shape representation 9
2.2. Statistical Shape Models 13
2.3. Establishing correspondence 26
2.4. Image segmentation with Active Shape Models 41
2.5. Alternatives to Active Shape Models 54
2.6. Quantification of segmentation accuracy 57
2.7. Further applications of shape models 59
2.8. Summary and discussion 59

3. Establishing surface correspondence **61**
3.1. Preliminaries 62
3.2. Establishing correspondence for shapes with spherical topology 63
3.3. Establishing correspondence for shapes with toroidal topology 77
3.4. Adaptive sampling in parameter space 83
3.5. Establishing correspondence for shapes with arbitrary topology 85
3.6. Improving pairwise methods using shape similarity trees 89

4. Evaluation of methods for establishing correspondence **93**
4.1. Correspondence evaluation by simulating image segmentation 93
4.2. Experimental setup 95
4.3. Comparison of shape similarity trees 99
4.4. Comparison of correspondence algorithms 101

5. The Probabilistic Active Shape Model **113**
- 5.1. Energy formulation . . . 114
- 5.2. Global shape energy . . . 115
- 5.3. Local shape energy . . . 118
- 5.4. Image energy . . . 120
- 5.5. Optimization . . . 121
- 5.6. Comparison to related approaches . . . 122

6. Applications **127**
- 6.1. Organ detection . . . 127
- 6.2. Liver segmentation . . . 132
- 6.3. Vertebra segmentation . . . 144
- 6.4. Prostate segmentation . . . 152

7. Conclusion **161**
- 7.1. Summary and discussion of contributions . . . 161
- 7.2. Future work . . . 163

A. Gradients **169**

B. Data sets and hardware **171**

C. Publications **173**

D. Supervising Activities **175**

E. Curriculum Vitae **177**

Glossary **179**

Bibliography **181**

1. Introduction

Imaging modalities like Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) are extremely important in modern medicine. These technologies are used to acquire 3D images of the human body, which allow physicians to look inside a patient, thus enabling them to make more accurate diagnoses and to optimize the patient's treatment. For example, diseases like cancer can often be diagnosed by identifying tumors in the acquired scans. Likewise, the subsequent treatment can be optimized with the exact knowledge of the patient's anatomy and the location of the tumor.

While in today's clinical practice, image-based diagnosis still depends mostly on the visual inspection of the tomographic scans by radiologists, the use of computers in clinics becomes more and more important. Interactive or automatic processing of medical images with computers can greatly enhance the clinical workflow and support physicians to make more accurate diagnoses. For example, the automatic extraction and subsequent visualization of anatomical structures can help surgeons to plan interventions such as liver resection [MTC02]. Similarly, exact knowledge about the size and position of a tumor facilitates patient specific radiation therapy, which allows to minimize the exposure of healthy tissue to radiation (e.g. [LCJ*12]). Another application is computer aided diagnosis, in which computers are used to estimate quantitative values reliably and reproducibly, such as for example the size of a lymph node [SBKW13]. In each of these examples, segmentation of anatomical structures in tomographic scans is crucial.

Image segmentation is the task of partitioning an image into several meaningful parts. In context of medical imaging, this usually means to label one or more organs in order to separate them from the background. An example is given in Figure 1.1, which shows the segmentation of a liver in a CT scan. Because manual segmentations of 3D images is a time-consuming process and hardly feasible in practice, there is a great need to automate the segmentation process. However, automatic segmentation of organs is far from being simple: Low contrast and blurry edges in medical images make it difficult to identify boundaries between different structures. Even for human observers, the interpretation of an image can be ambiguous, which is why manually created segmentations often exhibit a large intra- and inter-observer variability [WZW04].

In order to understand what makes segmentation of medical images challenging, one has to take a closer look at the image acquisition process. Image intensities are physical measurements, such as the tissue's ability to absorb X-rays in CT or its proton density in MRI. In many cases, tissues of adjacent organs have similar physical properties, and are therefore mapped to the same intensity range. This causes a low contrast between the organs, like in the image in Figure 1.2(a), that renders purely intensity-based segmentation impossible. For example, region-based segmentation methods such as region growing often fail because the segmentation 'leaks' from one organ into an adjacent structure

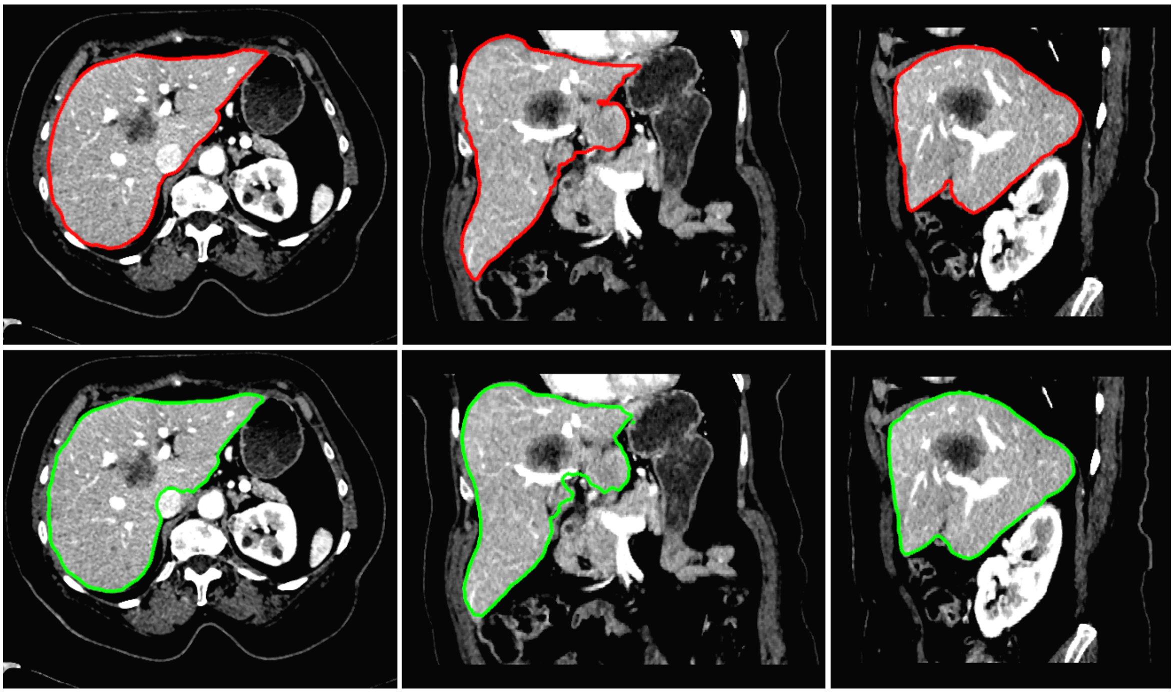
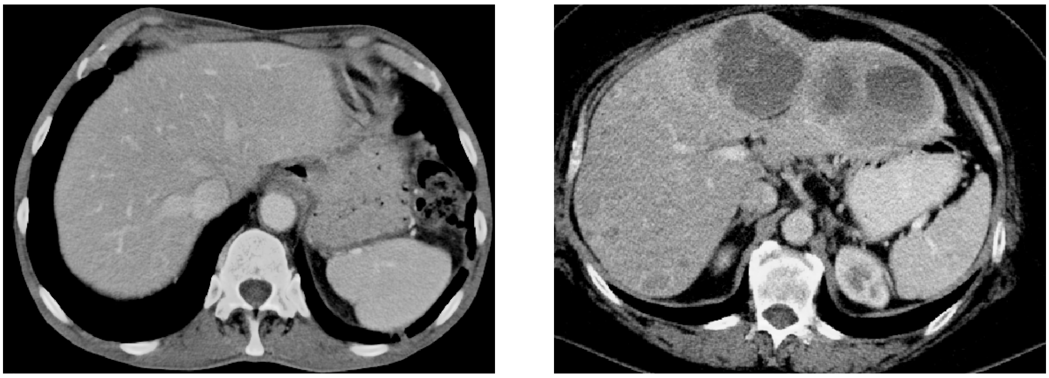


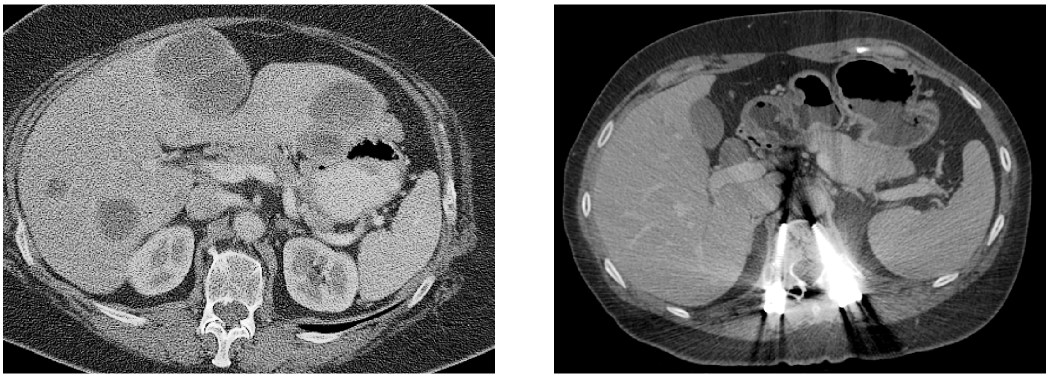
Figure 1.1.: Segmentation of the liver in a contrast-enhanced CT scan of the abdomen. Shown are, from left to right, a transversal, a coronal and a sagittal slice of the tomographic image. The segmentation is visualized as a contour that delineates the liver tissue. The top row shows a manual segmentation, and the bottom row an automatic segmentation computed with the Probabilistic Active Shape Model that is proposed in this thesis.

with similar appearance. Contrariwise, there is a risk that pathological tissue, as for example tumor tissue like in Figure 1.2(b), is not included into the segmentation, because it appears different than healthy tissue. Another problem with medical images is that their resolution is usually relatively low. This leads to partial volume effects, which are caused by averaging measurements of different tissues to a single intensity value. Partial volume effects make edges between adjacent structures appear blurry and impede intensity-based, region-based or edge-based segmentation. Additionally, the quality of medical images can be severely affected by noise (Figure 1.2(c)) or other image artifacts, such as metal shadows in CT images caused by implants (Figure 1.2(d)), or motion artifacts in MRI caused by breathing or some other movement of the patient.

Erdt et al. [ESS12] point out that the segmentation of an object in an image depends on three main characteristics: *object boundary*, *object homogeneity* and *object shape*. The first two characteristics are signal-based and thus strongly affected by the imaging modality. As discussed above, they often do not provide sufficient information for robust and exact segmentation. On the other hand, shape is an inherent property of the target structure itself and independent of the image acquisition process.



(a) Low contrast (b) Tumors



(c) Noise (d) Metal shadows

Figure 1.2.: Typical challenges in medical image segmentation: The segmentation of an organ, for example the liver, may be hampered by low contrast to adjacent organs such as the stomach (a) or inhomogeneous appearance of the tissue caused by tumors (b), noise (c) or image acquisition artifacts, such as for example metal shadows caused by pedicle screws (d).

Thus, *prior knowledge* about the object's shape provides additional constraints that make accurate segmentation feasible.

There are various different means to incorporate prior knowledge about the object's shape into the segmentation process. Many methods, for example Snakes [KWT88], use local shape priors that enforce that the segmentation boundary is smooth. Assuming smoothness is a reasonable assumption for organ boundaries, but this local prior is usually not sufficient to separate large adjacent structures with similar image appearance, such as for example liver and stomach in CT images. Because many organs have a characteristic shape, global shape priors are a powerful tool for segmentation. Global shape priors can be explicitly designed, for example by assuming that the organ is roughly ball-shaped, or statistically learned from a set of training examples.

Especially the latter kind of models, Statistical Shape Models (SSMs), have established themselves as a powerful and versatile tool for medical image segmentation. An SSM describes the typical shape of an organ as well as its major variations. During segmentation, the model is used to constrain the set of feasible shapes to those which are statistically plausible with respect to the patterns extracted from the training shapes. SSM-based approaches are among the most robust and accurate segmentation algorithms available for 3D organ segmentation. The best-performing approaches in several *Grand Challenges*, which are segmentation competitions co-located with the renowned *Medical Image Computing and Computer Assisted Intervention (MICCAI)* conference, are based on SSMs: Examples are competitions for segmentation of the liver in contrast-enhanced CT scans [HvGS*09], of knee cartilage [Ski] and of the prostate [Pro] in MRI scans. In contrast to competing model-based segmentation approaches, as for example atlas registration, SSM-based algorithms are usually very fast, which is especially important for their applicability in clinical practice.

Despite their successful use in medical image analysis, SSM-based algorithms have several problems and limitations. One of the key challenges is **model construction**: Before an SSM of an organ can be learned, it is necessary to establish correspondence between the training shapes. For each point on the surface of a training shape, the corresponding points on all other training examples have to be determined. Although solving this so-called *correspondence problem* has gained a lot of attention recently, the problem is far from being solved. Several approaches have been proposed that differ greatly in computational complexity and resulting model quality. Additionally, many approaches are only applicable to certain types of shape classes, such as classes with relatively low or moderate intra-class variability, or classes that only contain shapes with a specific topology, as for example shapes with spherical topology.

Another limitation of current SSM-based segmentation algorithms is that they are **too restrictive**: The segmentation of a previously unseen scan is usually restricted to shapes that are linear combinations of the training shapes. While this restrictive strategy makes SSM-based segmentation very robust, it also impedes a high segmentation accuracy, because one can not expect that every possible shape instance of an organ has been previously observed in the training shapes. Such previously unseen shapes can often not be reproduced accurately by the SSM. The huge shape variability of many organs and the presence of pathologies that can cause abnormal variations from the ‘normal’ shape make it unlikely that an SSM can be learned that accounts for every possible variation. The problem is enforced by the fact that in medical imaging, usually only few training examples are available, because obtaining the necessary manual segmentations to generate new examples is costly. Even if many training examples were available, it is often reasonable to reduce the dimensionality of the SSM to the most frequent shape variations in order to decrease the computation time. Due to the lack of time and of sophisticated hardware in clinics, computationally complex models are of limited use.

Several approaches have been proposed in the literature in order to increase the flexibility of SSMs (e.g. [WKL*01, EKSW10]), or to post-process segmentation results obtained by SSM-based segmentation (e.g. [KLL07]). However, these approaches are usually hybrid approaches that introduce additional model assumptions and are governed by several free parameters, which make the adaption of the segmentation algorithm to different applications harder.

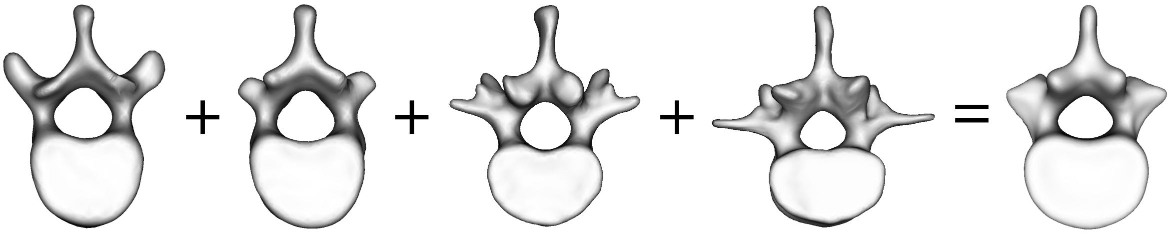


Figure 1.3.: Is the assumption that shapes are linear combinations always reasonable? The figure shows from left to right two thoracic vertebrae, two lumbar vertebrae and a mean shape computed from both vertebra types. The mean shape shows characteristics of both thoracic and vertebrae types, but does not correspond to a vertebra shape that one would expect to find in the human body. Nevertheless, the mean shape is the most probable shape in a linear shape model.

Finally, the vast majority of SSM-based approaches applied to medical images use linear shape models. Although linear shape models have shown good results in many applications, the underlying model assumption that plausible shapes are linear combinations of the training shapes is not necessarily true. This is illustrated in Figure 1.3, which shows examples of lumbar and thoracic vertebrae, as well as a mean shape computed from both vertebra types. Although the mean shape is the most probable shape under the assumption of a multivariate Gaussian shape distribution, one would not expect to find such a vertebra in the human body, because it neither corresponds to a lumbar, nor to a thoracic vertebra. In this example, the linear model is **too unspecific** because it describes a broader class of shapes than desired. More sophisticated, non-linear machine learning methods can be used to generate more specific SSMs in such applications. Although it can be argued that in the example given in Figure 1.3, one could have used separate models for different vertebra types, the use of nonlinear techniques to learn models that contain different classes of similar shape potentially allows to segment and identify objects simultaneously. Up to now, nonlinear SSMs have hardly attracted attention in medical imaging so far. The few approaches that exist do not provide a general framework that allows seamless integration of both linear or nonlinear SSMs.

1.1. Contributions

This thesis contributes several new algorithms for shape model-based image segmentation that address the challenges mentioned in the introduction. For **model construction**, various new solutions are devised that solve the correspondence problem for shapes of different topologies and shape complexity. Furthermore, a new image segmentation algorithm is proposed, the *Probabilistic Active Shape Model*, which facilitates image segmentation with **flexible** and **specific** shape priors. In particular, this thesis has the following main contributions:

- It proposes Parameter Space Propagation, a new method for consistent spherical parameterization of groups of shapes. This methods can either be used to directly construct shape models, or provide a robust initialization for subsequent groupwise correspondence optimization.
- It describes the first approach for groupwise optimization by reparameterization for shapes of genus 1 topology. The algorithm consistently parameterizes shapes of genus 1 to a planar parameter domain, and subsequently optimizes the correspondence with a reparameterization function dedicated to the parameter domain.
- It introduces an algorithm for constructing adaptively sampled shape models from different parameter domains, which allows for example to construct shape models that use more landmarks in regions with high curvature.
- It proposes a new multiscale algorithm for nonrigid registration of shapes, and describes how this algorithm can be used to solve the correspondence problem.
- It provides a detailed evaluation of the proposed correspondence algorithms on liver, prostate, lumbar vertebra and thoracic vertebra shapes. This evaluation includes an analysis of combining pairwise correspondence algorithms with shape similarity trees. It also includes a comparison of the proposed methods with a state-of-the-art groupwise optimization algorithm. Moreover, a new segmentation-based evaluation methodology is proposed that allows to quantify the impact of correspondence on the segmentation accuracy.
- It describes the Probabilistic Active Shape Model, a new shape model-based image segmentation algorithm. This algorithm unifies the treatment of linear and nonlinear shape priors, and allows to use these priors in a flexible, non-restrictive way, while still being mathematically concise. The global shape prior is enriched with a local model which is also statistically learned. Thus, the algorithm remains easily adaptable to different applications.
- It provides a detailed evaluation of the Probabilistic Active Shape Model by applying it to three real world image segmentation problems: liver and vertebra segmentation in CT scans and prostate segmentation in T2-weighted MRI scans.

1.2. Structure of this thesis and notation

This thesis is structured as follows: Chapter 2 reviews the state-of-the-art of shape model-based medical image segmentation. It covers algorithms for establishing correspondence, methods for modeling shape variation, image segmentation algorithms such as the Active Shape Model and evaluation methodology for assessing the quality of image segmentation. Chapter 3 describes new algorithms for establishing correspondence between shapes that enable the construction of shape models for different shape classes. It describes methods for consistent parameterization of shapes of genus 0 and of genus 1 topology, and discusses groupwise correspondence optimization based on these parameterizations. It also treats a topology-independent algorithm for constructing shape models as well as means for improving pairwise correspondence methods. A detailed evaluation of these methods is given in Chapter 4. Chapter 5 describes a new algorithm for shape model-based image segmentation, the Probabilistic Active Shape Model, which is more flexible than the standard Active Shape Model and is able to incorporate nonlinear shape priors. The Probabilistic Active Shape Model is evaluated in Chapter 6 by applying it to several real world segmentation problems: Liver and vertebra segmentation in CT images and prostate segmentation in MRI scans. Finally, Chapter 7 concludes the thesis by providing a summary and discussion of its contributions. It also recommends direction for future work and new research ideas based on this work.

Appendix A lists the gradients of the energies used in the Probabilistic Active Shape Models. Appendix B provides a detailed description of the data sets used throughout this thesis. Some parts of this thesis have been published in peer-reviewed conference proceedings and as journal publications. A list of these publications is given Appendix C. The work has also been influenced by several Diploma, Master's and Bachelor's theses that have been written under my supervision by students graduating at Technische Universität Darmstadt. These theses are listed in Appendix D. Finally, Appendix E gives a short curriculum vitae that covers my education and work experience.

Throughout this thesis, the following notation for mathematical symbols is used in order to ease readability and understanding of formulas. Small letters are used for constants, variables and vectors. Additionally, vectors are printed in bold, but not their elements, as in $\boldsymbol{x} = \{x_1, \ldots, x_{3n}\}$. Large letters are used for tuples that contain different elements, such as meshes or graphs, and for matrices. The latter are printed in bold. Sets are denoted using calligraphic letters like $\mathcal{X}$.

Many letters are only used for a single purpose in the whole thesis. For example, n denotes the number of landmarks of a shape. A list of these symbols is compiled in the glossary which can be found on page 179.

2. State of the art

The use of shape priors is nowadays a standard technique in medical image analysis. This chapter introduces the reader to the current state of the art of shape modeling, covering aspects such as shape representation, establishing correspondence, model learning and the use of shape models for medical image segmentation. The emphasis is laid on methods closely related to the techniques devised in this thesis, but some alternative modeling choices are reviewed as well.

The chapter is structured as follows: Section 2.1 introduces different shape representations, including the landmark representation used within this thesis. Section 2.2 shows how the variability of shape of an object class can be described with linear or nonlinear models. Additionally, methods for modeling the joint variability of several shapes with coupled and articulated models are briefly outlined. A prerequisite for model learning is to identify corresponding landmarks on a set of shapes. Methods for establishing correspondence are reviewed in Section 2.3. Section 2.4 covers the Active Shape Model (ASM), an image segmentation algorithm using shape priors, which is extended in this work. Alternative shape model-based segmentation methods are discussed in Section 2.5. An important aspect in medical image segmentation is quantitative evaluation of segmentation accuracy. To this end, several volumetric and surface-based measures to quantify the accuracy are presented in Section 2.6. Additional applications of SSMs beyond image segmentation, such as shape extrapolation and shape analysis, are briefly outlined in Section 2.7. The chapter concludes in Section 2.8 with a summary and discussion of limitations of the current state of the art.

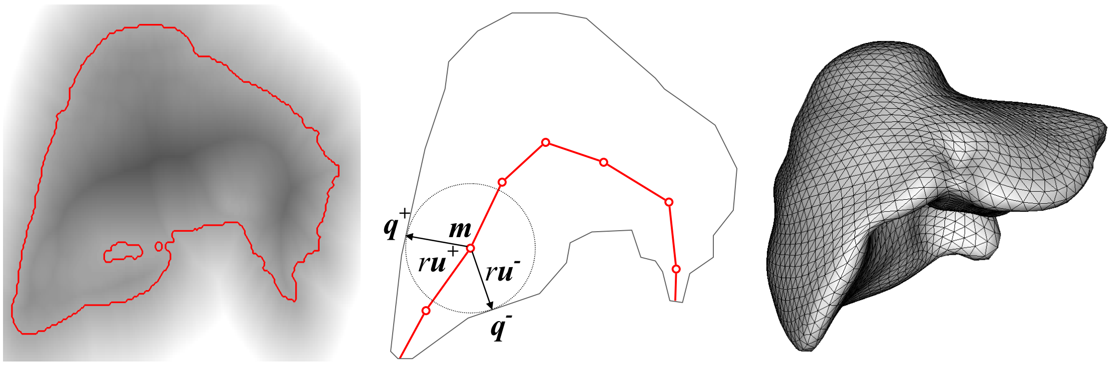
2.1. Shape representation

This section discusses three popular shape representations which can be used for segmentation: Implicit functions (Section 2.1.1), m-reps (Section 2.1.2) and landmarks (Section 2.1.3). Figure 2.1 shows the shape of a liver represented with each of the modeling techniques. Advantages and disadvantages of the different shape representations are discussed in Section 2.1.4. It is assumed that a shape is a $(d-1)$-dimensional surface embedded in a d-dimensional space.

2.1.1. Implicit shape representation

In implicit shape representations, the contour $\mathcal{C}$ of a shape corresponds to the *zero level set* of an embedding function $\zeta : \mathbb{R}^d \rightarrow \mathbb{R}$, that is

$$\mathcal{C} = \{\boldsymbol{z} \in \mathbb{R}^d \mid \zeta(\boldsymbol{z}) = 0\}. \tag{2.1}$$



(a) Implicit representation (b) m-rep (c) Landmark representation

Figure 2.1.: Different representations of a liver shape. (a): Slice of a 3D signed distance map with zero level set (red contour). Larger function values appear brighter. (b): Illustration of an m-rep mesh with seven medial atoms. The circle shows a maximum inscribed ball around one atom. (c): Triangulated mesh with 2562 landmarks.

Examples for the embedding function $\zeta(\boldsymbol{z})$ are signed distance maps (SDMs), which map a point $\boldsymbol{z} \in \mathbb{R}^d$ to the positive or negative distance between $\boldsymbol{z}$ and its closest point on the shape's surface. Negative function values indicate that $\boldsymbol{z}$ is within the shape. By definition, the zero level set of $\zeta(\boldsymbol{z})$ corresponds to the points on the shape's contour. An illustration of a SDM is shown in Figure 2.1(a).

Implicit shape representations are used in level set segmentation, which has been pioneered by Osher and Sethian [OS88]. The basic principle is to minimize an energy functional by evolving the zero level set over time. The functional typically consists of external forces that draw the contour towards image features such as strong gradients, and internal forces which maintain certain properties of the contour, such as smoothness or statistical probability. For more information on level set segmentation, the reader is referred to the review of Cremers et al. [CRD07].

Leventon et al. [LGF00] show how Principal Component Analysis (PCA) can be used to incorporate global statistical shape information into the level set segmentation framework. Each training shape is represented by a separate SDM. Note that the SDMs can be easily aligned by superimposing the training shapes. For a discrete representation, the continuous SDMs are sampled on a regular grid that is large enough to contain the zero level sets of all shapes. The grids are represented as high-dimensional vectors in order to facilitate statistical learning with PCA.

Applying PCA to learn statistics of SDMs is methodologically flawed, because PCA assumes that the training examples are elements of a linear vector space, which is not the case for SDMs [CRD07]. On the other hand, the modeling technique has been frequently used and seems to work well in practical applications. Mathematically more profound learning methods can be realized by mapping SDMs into the 'LogOdds' space [PFB*07], or by using kernel techniques for nonparametric density estimation to capture the shape variation [RC05, DRT08, WSH09]. Level set segmentation with shape priors has been frequently applied to 3D segmentation of medical images, for example for liver [WSH09] and prostate segmentation [TM12].

2.1.2. M-reps

M-reps [SP08] are a medial object representation which is based on the concept of the *medial locus* of an object. Suppose that the object is a closed subset $\mathcal{O} \subset \mathbb{R}^d$. The medial locus of $\mathcal{O}$ is the set $\{(\boldsymbol{o}, r)\}$ of center points $\boldsymbol{o}$ and corresponding radii r of the *maximal inscribed balls* of $\mathcal{O}$. Maximal inscribed balls are balls $\mathcal{B} \subset \mathcal{O}$ such that there is no other ball $\mathcal{B}' \subset \mathcal{O}$ with $\mathcal{B} \subset \mathcal{B}'$.

The basic idea of m-reps is to approximate the medial locus of an object by a mesh which consists of a finite set of medial atoms. A d-dimensional medial atom[1] is a tuple $(\boldsymbol{m}, r, \boldsymbol{u}^+, \boldsymbol{u}^-)$, where

- $\boldsymbol{m} \in \mathbb{R}^d$ is the position of the atom on the medial locus,
- $r \in \mathbb{R}_+$ is the radius of the maximal inscribed ball at $\boldsymbol{m}$, that is, the distance of $\boldsymbol{m}$ to the object's surface,
- $\boldsymbol{u}^+, \boldsymbol{u}^- \in \mathbb{R}^d$ are unit length vectors, the so-called *spoke orientations*.

The points $\boldsymbol{q}^+, \boldsymbol{q}^- \in \mathbb{R}^d$, with $\boldsymbol{q}^\pm = \boldsymbol{m} + r\boldsymbol{u}^\pm$ are the implied boundary points of $\boldsymbol{m}$. Figure 2.1(b) shows a 2D m-rep with a detailed illustration of one of its medial atoms.

An arbitrary element on the medial locus of $\mathcal{S}$ is approximated by linear interpolation of two or more medial atoms that form a face on the m-rep mesh. Likewise, the implied boundary points of the interpolated medial locus element can be computed. This means that the m-rep serves as a parameterization of the surface of $\mathcal{O}$.

In order to describe complex objects with protrusions and indentations, m-reps are decomposed into a hierarchy of several meshes of medial atoms, so-called figures, and interrelations between these figures can be defined. In general, the same object may be described by several figural decompositions [PFY*99]: For example, a hand can be modeled by a main figure for the palm, and five protrusions for the fingers. Alternatively, palms and fingers may be described by a single main figure, and the gaps between the fingers by several intrusions.

Deformable m-reps [PFJ*03] have been used for the segmentation of organs in 3D medical images, where the objective function was either geometrically or statistically defined. Statistically learned shape models based on m-reps are learned from a set of training examples. Statistical model learning requires that correspondence between shapes must be established, that is, each shape must be represented by nets of medial atoms which have the same topology. A problem here is that medial axis representations are unstable with respect to small changes of the boundary of an object: In fact, it was shown by August et al. [ATZ99] that even smoothing the boundary may lead to new branches in the medial representation. An approach for automatic construction of corresponding m-reps for a set of biological shapes has been proposed by Styner et al. [SGJP03].

---

[1] Strictly speaking, this is the definition of a medial atom of order 1. The reader is referred to Siddiqi and Pizer [SP08] for more details.

2.1.3. Landmark representation

In this thesis, an explicit shape representation (e.g. [CTCG95]) is used in which each shape is approximated by n points $\boldsymbol{x}^j \in \mathbb{R}^d$, $j \in \{1, \ldots, n\}$. These points are called *landmarks*, because each of them is associated with a specific anatomical location. The *landmark vector* of a d-dimensional shape is obtained by concatenating its landmarks:

$$\boldsymbol{x} = (x_1^1, \ldots, x_d^1, \ldots, x_1^n, \ldots, x_d^n) \in \mathbb{R}^{dn}. \tag{2.2}$$

In order to describe the whole contour of a shape, and not only landmarks, a connectivity can be defined that describes how landmarks are connected. For 3D shapes, the connectivity is usually a triangulation. Figure 2.1(c) shows a 3D shape in landmark representation, which is basically a triangulated mesh, where the mesh points are the landmarks. Since the connectivity is fixed for all shapes of a particular class, it is in most cases sufficient to specify the landmark vector of a shape. Whenever the connectivity is important, $M(\boldsymbol{x})$ is used to denote the mesh corresponding to the landmark vector $\boldsymbol{x}$.

In order to learn an SSM from a set of training shapes in landmark representation, each example must be represented with the same number of landmarks. Moreover, the shapes must be in *index correspondence*: For two shapes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and all $k \in \{1, \ldots, n\}$, the landmarks $\boldsymbol{x}_i^k$ and $\boldsymbol{x}_j^k$ must represent corresponding points on the shapes' surfaces. Both requirements are crucial for the learning process: The first ensures that the shapes are elements of the same vector space, whereas the second enables the learning algorithm to extract meaningful patterns from the data. Establishing correspondence between training shapes is a challenging problem and still subject to active research. Methods for establishing correspondence are reviewed in Section 2.3. Methods for learning SSMs from training shapes in landmark representation are discussed in Section 2.2.

2.1.4. Discussion of different shape representations

In context of medical image segmentation, the landmark representation has several advantages over implicit representations and m-reps. Landmarks provide a very intuitive shape representation that can be easily understood. Although 3D landmark vectors have a high dimensionality, typically in the order of 10000, they provide a relatively sparse representation when compared to discretely sampled SDMs. In contrast to SDMs, landmark vectors form a linear vector space such that linear machine learning techniques like PCA are directly applicable. Segmentation algorithms using the landmark representations are usually faster than for example level set algorithms. Because landmarks encode anatomical positions, localized information about the image appearance or shape can be easily incorporated into the algorithms. Segmentation algorithms based on landmarks achieved the best results in several segmentation challenges such as the *Sliver07* challenge for liver segmentation in CT scans [HvGS*09], the *Ski10* challenge for cartilage and bone segmentation in knee MRI scans [Ski], and the PROMISE12 challenge for prostate segmentation in MRI scans [Pro].

The greatest drawback of landmarks is that correspondence between shapes must be established. Solving the correspondence problem is commonly regarded as the most challenging problem that must be encountered when choosing this representation. In implicit shape representation, a rough alignment of the shapes is used to superimpose the SDMs. Leventon et al. [LGF00] report that signed distance maps are relatively robust to small misalignment. On the other hand, the demand for exact correspondences of landmarks might be advantageous when modeling local information.

Implicit representations are very flexible and versatile. For example, level set segmentation allows a change of topology during segmentation. Local shape information which is not restricted to a particular anatomical region, such as smoothness of the contour, can be easily modeled. These two properties make level sets also applicable to amorphous structures whose shape can be best described by local shape properties, but not by global constraints. In contrast to that, the landmark representation does not per se allow for the change of topology, and the association of landmarks with particular anatomical regions restricts their application to structures that have a distinctive global shape.

Relatively few authors have chosen m-reps for shape modeling so far. Similar to landmarks, correspondence must be established in order to capture statistical information. However, this task appears to be extremely complex, partly because of the modeling ambiguities when using m-reps, partly because of the instability of the representation when the shape of an object changes. These problems, and the lack of solutions proposed so far, are strong arguments against the choice of m-reps for image segmentation.

2.2. Statistical Shape Models

Statistical Shape Models (SSMs) are a statistical description of the variability of shape of an object class which is learned from a set of training examples. Although the term SSM is sometimes used for other shape representations as well, it refers in this work always to models learned from landmark vectors. Especially in older publications, landmark-based SSMs are also called *Point Distribution Models*.

The set of training shapes is denoted by $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s\}$, where $\boldsymbol{x}_i \in \mathbb{R}^{dn}$, $i \in \{1, \ldots, s\}$, is a d-dimensional shape with n landmarks. An SSM is learned from $\mathcal{X}$ in two steps: Firstly, the training shapes are aligned into a common coordinate system using *Procrustes Alignment* in order to remove the influence of scaling, rotation and translation. Procrustes alignment is discussed in Section 2.2.1. Secondly, the SSM is computed by applying a learning algorithm to the aligned shapes. In most cases, PCA is used to learn a linear shape model, which is discussed in Section 2.2.2. Section 2.2.3 presents shape models which use Independent Component Analysis (ICA) instead of PCA to capture the shape variability. While at least some researchers applied ICA-based shape modeling to medical image data, nonlinear techniques such as Gaussian Mixture Modeling or Kernel PCA got hardly any attention in the community at all. Almost all nonlinear techniques reviewed in Section 2.2.4 have only been applied to small 2D data sets or toy examples. In contrast to that, coupled shape models have gained some popularity and become increasingly important. Coupled models describe the joint variability of a set of structures, for example of adjacent organs, or the variability of structures that can be decomposed into several parts. An interesting subclass of coupled models are articulated models which are used for example to model joints such as the knee. Coupled and articulated shape models are discussed in Section 2.2.5.

2.2.1. Aligning Shapes

According to the common definition, the shape of an object is invariant to scale, rotation and translation [Goo91]. Since the size of an organ differs from patient to patient, the pose of patients within the tomographic scanner may differ, and because scanners may generate images with different coordinate systems, training shapes extracted from medical images may vary significantly in their pose. Before the shape variation of an organ can be learned, these differences in scale, rotation and translation must be removed from the training shapes. The common approach to do this is called Generalized Procrustes analysis (GPA) [Gow75].

Cootes and Taylor [CT04] describe a simple iterative variant of GPA for aligning a set of shapes $\mathcal{X}$. It aligns the shapes such that the sum of squared distance $\sum_{i=1}^{s} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|^2$ to the mean $\bar{\boldsymbol{x}} = \frac{1}{s}\sum_{i=1}^{s} \boldsymbol{x}_i$ is minimized. As this is an underconstrained problem, additional constraints on scale and rotation are imposed.

2.2.2. Linear shape model

A linear SSM is learned by applying PCA to the aligned training shapes. PCA is a linear method that finds a low dimensional representation of the training data. Let

$$\bar{\boldsymbol{x}} = \sum_{i=1}^{s} \boldsymbol{x}_i \tag{2.3}$$

denote the mean shape, $\boldsymbol{X} = (\boldsymbol{x}_1 - \bar{\boldsymbol{x}} \mid \ldots \mid \boldsymbol{x}_s - \bar{\boldsymbol{x}}) \in \mathbb{R}^{dn \times s}$ denote a matrix in which each column is a centered training vector, and let

$$\boldsymbol{\Sigma} = \frac{1}{s-1} \boldsymbol{X}\boldsymbol{X}^T \tag{2.4}$$

denote the sample covariance matrix of the shapes. A PCA is done by computing the eigenvectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{dn}$ and corresponding eigenvalues $\lambda_1 \geq \ldots \geq \lambda_{dn} \geq 0$ of $\boldsymbol{\Sigma}$. In context of shape modeling, the eigenvectors are frequently called *modes of variation* or simply *modes*. An illustration of the first mode of variation of a linear shape model of the pancreas is shown in Figure 2.2.

An eigenvalue λ_i measures the variance of the training data when projected on the corresponding eigenvector $\boldsymbol{p}_i$. Therefore, eigenvalues measure in some sense the importance of eigenvectors. Dimensionality reduction is achieved by discarding eigenvectors with small or zero eigenvalue. The remaining t principal eigenvectors are combined to a matrix $\boldsymbol{P} = (\boldsymbol{p}_1 \mid \ldots \mid \boldsymbol{p}_t) \in \mathbb{R}^{dn \times t}$. The most frequently employed approach to select t for SSMs is to retain eigenvectors that explain a certain percentage of the variance, for example 98 %. This means one selects the largest t such that $\sum_{i=1}^{t} \lambda_i / \sum_{i=1}^{dn} \lambda_i > 0.98$. Mei et al. [MFR*08] propose a more complex approach for determining the dimensionality: They use bootstrapping in order to estimate how stable a mode of variation is, and discard unstable modes.

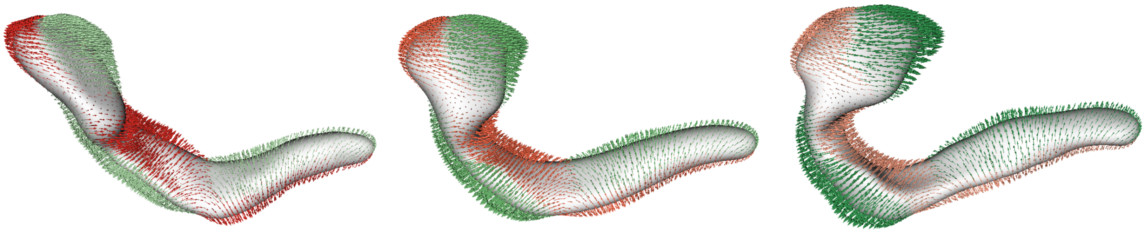


Figure 2.2.: Visualization of the first mode of variation of a linear pancreas shape model. The shapes shown are from left to right: $\bar{\boldsymbol{x}} - 2\sqrt{\lambda_1}\boldsymbol{p}_1$, the mean shape $\bar{\boldsymbol{x}}$, and $\bar{\boldsymbol{x}} + 2\sqrt{\lambda_1}\boldsymbol{p}_1$. The arrows visualize how landmarks move when the first mode $\boldsymbol{p}_1$ is weighted. They move along the green arrows for positive weights and in the opposite direction (red arrows) for negative weights. This model has been constructed using pairwise nonrigid registration (Section 3.5.1.)

The mean shape $\bar{\boldsymbol{x}}$ and the matrix $\boldsymbol{P}$ allow to represent each training shape by

$$\boldsymbol{x}_i = \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b}_i + \boldsymbol{r}_i \tag{2.5}$$

where $\boldsymbol{b}_i \in \mathbb{R}^t$ is a low-dimensional vector, which is called the shape's *parameter vector* in this work. By dropping the residual vector $\boldsymbol{r}_i \in \mathbb{R}^{dn}$, the approximation

$$\boldsymbol{x}_i \approx \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b}_i \tag{2.6}$$

is obtained, which allows to represent a shape by its low-dimensional parameter vector only. The elements of the set

$$\left\{\bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b} \mid \boldsymbol{b} \in \mathbb{R}^t\right\} \tag{2.7}$$

constitute the *principal subspace*, which is a t-dimensional affine subspace of $\mathbb{R}^{dn}$. The size of t is bounded by the number of training examples, in particular $t \leq s-1$. This is because the training shapes span at most an s dimensional vector space. An additional dimension 'gets lost' by subtracting the mean when constructing $\boldsymbol{X}$. Therefore, the number of nonzero eigenvalues of $\boldsymbol{\Sigma}$ can be at most $s-1$.

One view on PCA is that it finds the low-dimensional, orthonormal coordinate system which best describes the data. For a given training set $\mathcal{X}$, it minimizes the error $\sum_{i=1}^{s} \|\boldsymbol{r}_i\|^2$ of the low-dimensional approximation in Equation 2.6. An alternative, but equivalent view is that it tries to find orthonormal axes on which the variance of the training set is maximized. Detailed explanation of these views on PCA can be found in the textbook of Bishop [Bis06].

2.2.2.1. Gaussian shape distribution

PCA is also closely related to the multivariate Gaussian distribution: Assuming for a moment that $\boldsymbol{\Sigma}$ has full rank, the distribution of the training shapes can be modeled by the multivariate Gaussian

$$N(\boldsymbol{x}; \bar{\boldsymbol{x}}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{dn}{2}} \det \boldsymbol{\Sigma}^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\bar{\boldsymbol{x}})\right) \tag{2.8}$$

with mean $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{\Sigma}$. If all eigenvectors are retained after diagonalizing $\boldsymbol{\Sigma}$, that is, $t = dn$, Equation 2.8 can be factored into[2]

$$\mathrm{N}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma}) = \prod_{i=1}^{dn} \frac{\exp\left(-\frac{b_i^2}{2\lambda_i}\right)}{(2\pi\lambda_i)^{\frac{1}{2}}}. \tag{2.9}$$

This means that the multivariate Gaussian can be factored into the product of one-dimensional Gaussians along the eigenvectors $\boldsymbol{p}_1,\dots,\boldsymbol{p}_{dn}$ of $\boldsymbol{\Sigma}$.

Cremers et al. [CTWS02] assess the probability of shapes with the Mahalanobis Distance

$$\mathrm{M}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma}) = \frac{1}{2}\sum_{i=1}^{dn} \frac{b_i^2}{\lambda_i}, \tag{2.10}$$

which corresponds to the negative log-likelihood of the Gaussian in Equation 2.8 without some constant terms. Even for the 2D shapes considered by Cremers et al., the problem easily occurs that too few training examples are available. As the rank of $\boldsymbol{\Sigma}$ is equal to the number of nonzero eigenvalues and thus is at most $s-1$, the matrix $\boldsymbol{\Sigma}$ is often singular in practice. Cremers et al. solve this by replacing all nonzero eigenvalues by a regularization constant λ'. They set $\lambda' = \frac{\lambda_r}{2}$, with λ_r being the smallest non-zero eigenvalue.

This approach is related to a more general machine learning technique proposed by Moghaddam and Pentland [MP97] which can be used to obtain a sparse representation of a high dimensional multivariate Gaussian. Assuming now that $t < dn$, Equation 2.8 can be approximated by the product of two marginal and independent Gaussians

$$\hat{\mathrm{N}}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma},\rho) = \left(\prod_{i=1}^{t} \frac{\exp\left(-\frac{b_i^2}{2\lambda_i}\right)}{(2\pi\lambda_i)^{\frac{1}{2}}}\right) \cdot \left(\frac{\exp\left(-\frac{\|\boldsymbol{r}\|^2}{2\rho}\right)}{(2\pi\rho)^{\frac{dn-t}{2}}}\right) \tag{2.11}$$

and the Mahalanobis distance becomes

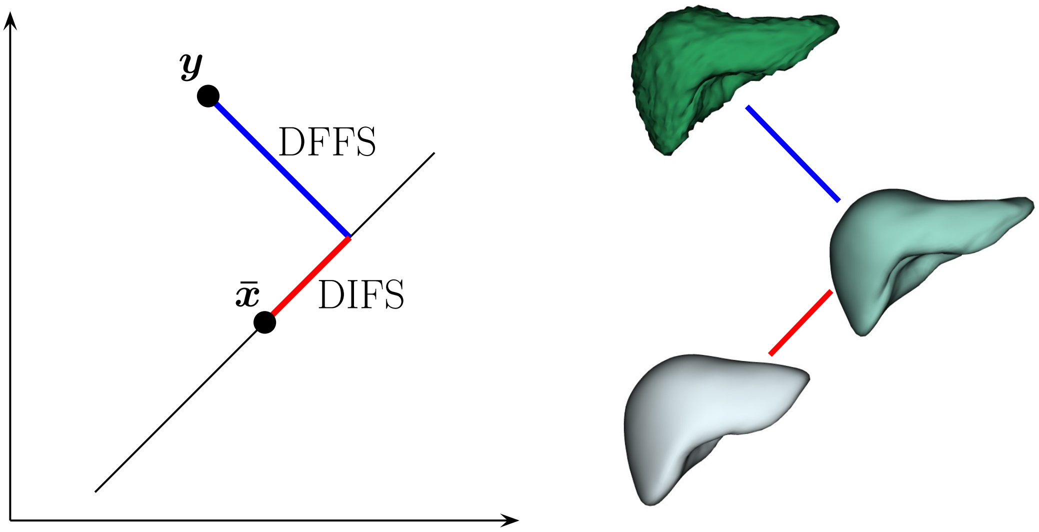
$$\hat{\mathrm{M}}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma},\rho) = \frac{1}{2}\sum_{i=1}^{t} \frac{b_i^2}{\lambda_i} + \frac{1}{2\rho}\|\boldsymbol{r}\|^2. \tag{2.12}$$

For $\rho = \frac{\lambda_r}{2}$, the regularized Mahalanobis distance used by Cremers et al. [CTWS02] is identical to the approximated Mahalanobis distance in Equation 2.12. Moghaddam and Pentland show that from an information theoretical point of view,

$$\rho = \frac{\sum_{i=t+1}^{dn} \lambda_i}{dn - t} \tag{2.13}$$

---

[2] Please refer to the textbook of Bishop [Bis06] (pages 78–86) or Moghaddam and Pentland [MP97] for a detailed explanation and derivation.



(a) DIFS and DFFS for a 1d feature space (b) DIFS and DFFS for a linear shape space

Figure 2.3.: Illustration of the *distance in feature space* (DIFS; red lines) and the *distance from feature space* (DFFS; blue lines). (a): Visualization using a 2D input space and a 1D feature space (black line). The DFFS equals the costs of projecting $\boldsymbol{y}$ to the feature space, and the DIFS is the distance of the projected point to the mean $\bar{\boldsymbol{x}}$. (b): For linear shape spaces, the DFFS equals the costs of projecting an arbitrary shape (shown highly saturated) to the nearest model representation (medium saturation), and the DIFS is the statistical distance of the model representation to the model's mean shape (low saturation).

is the optimal choice because it minimizes the Kullback-Leibler divergence

$$\mathrm{KL}(\rho) = \mathrm{E}\left[\frac{\mathrm{N}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma})}{\hat{\mathrm{N}}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma},\rho)}\right], \tag{2.14}$$

with $\mathrm{E}[\cdot]$ being the expected value. The advantage of this approximation is that one only needs the first t principal eigenvectors for evaluating $\hat{\mathrm{M}}(\boldsymbol{x};\bar{\boldsymbol{x}},\boldsymbol{\Sigma},\rho)$, because it is $\|\boldsymbol{r}\|^2 = \|\boldsymbol{x}-\bar{\boldsymbol{x}}\|^2 - \|\boldsymbol{b}\|^2$. It therefore allows to estimate a high-dimensional probability distribution while still using dimensionality reduction.

Moghaddam and Pentland provide an abstract point of view of their approximation, which is not restricted to Gaussians: By mapping high-dimensional data to an abstract low-dimensional feature space, two kinds of costs can be considered: The *distance in feature space* (DIFS) and the *distance from feature space* (DFFS). The DIFS is for example a density in feature space that has been learned from training data. On the other hand, the DFFS accounts for the loss of information that occurs when projecting high dimensional data to a low dimensional space. An illustration of the concept is shown in Figure 2.3. In case of the approximation of the multivariate Gaussian, the feature space corresponds to the linear vector space $\mathbb{R}^t$ spanned by the first t principal axes. The DIFS is the Mahalanobis distance within this subspace, and the DFFS equals the costs of the projection to the subspace. Moghaddam and Pentland's technique is applicable to other feature spaces as well, for example those learned from Kernel PCA. It is later used for both the linear and the nonlinear shape energy of the Probabilistic Active Shape Model (Chapter 5), which is one of the contributions of this thesis.

2.2.2.2. Learning linear models

Algorithmically, the most time consuming part of PCA is the diagonalization of the covariance matrix $\boldsymbol{\Sigma}$. The straightforward approach is to apply a diagonalization algorithm directly to $\boldsymbol{\Sigma}$. Because the covariance matrix has millions of entries for 3D shapes with a few thousand landmarks, diagonalization of $\boldsymbol{\Sigma}$ can easily take several minutes on modern desktop computers. In case that there are considerably less training examples than dimensions ($s \ll dn$), which is usually the case in shape modeling, a faster technique is to use dual PCA [CT04]. Instead of diagonalizing the $dn \times dn$-matrix $\boldsymbol{\Sigma}$, one diagonalizes the $s \times s$-matrix $\hat{\boldsymbol{\Sigma}} = (s-1)^{-1} \boldsymbol{X}^T \boldsymbol{X}$. One can show that $\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\Sigma}}$ have identical non-zero eigenvalues. Likewise, one can compute the corresponding eigenvectors of $\boldsymbol{\Sigma}$ from the eigenvectors of $\hat{\boldsymbol{\Sigma}}$. A second alternative for performing PCA is to compute the Singular Value Decomposition of $\boldsymbol{X}$ [Kal96]. This is advantageous because the matrices $\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\Sigma}}$ are ill-conditioned in a numerical sense. Finally, for learning shape models with outliers or corrupted data, Lüthi et al. [LAV09] propose an approach that is based on Probabilistic PCA.

2.2.2.3. Increasing the flexibility of linear shape models

As mentioned above, the dimension of the principal subspace learned from the training shapes is bounded by $s-1$. In case of few training samples, this subspace is often too small to allow for a good generalization of the SSM to unseen shapes. Therefore, several techniques have been proposed to increase the flexibility of the model. It should be noted that for image segmentation, the techniques for increasing the flexibility discussed here are complementary to approaches that try to impose shape constraints in a flexible way. Such techniques are reviewed in Section 2.4.4.

One can distinguish between three different strategies for increasing the SSMs flexibility:

1. Build a hierarchical shape model that decomposes the shape into several subparts or frequencies, and do a PCA of these parts.
2. Increase the number of training shapes by synthesizing artificial shapes from the existing examples.
3. Directly incorporate additional variation into the sample covariance matrix.

The intuition why hierarchical models increase the flexibility is that the shape space spanned by the combination of several learned localized variations is larger than the global shape space. Suppose that a shape is decomposed into several parts or *subshapes*, such that each subshape consists of several landmarks. If a shape model for each subshape is computed individually, the dimension of each submodel is again bounded by $s-1$. However, if a subshape is sampled from each submodel and all subshapes are recombined to a single shape, a larger class of shapes can be generated than with a single SSM, because constraints imposed by global correlations are ignored. On the downside, this approach may also affect the model's specificity. The important point is therefore to determine reasonable decompositions and recombine them sensibly.

Davatzikos et al. [DTS03] use the wavelet transform to decompose shapes into 64 bands. Bands with low index represent global, low frequency shape information, whereas bands with high index encode more local, high frequency shape information. PCA is then performed on the covariance matrix of each band, instead of doing only a single PCA on the shape covariance matrix. The hierarchical SSM showed superior performance compared to a standard SSM during image segmentation. However, the results of the study indicate that the gap between hierarchical SSM and linear SSM decreases with increasing number of training samples.

Another hierarchical shape model has been proposed by Zhao et al. [ZAT05], who decompose the model mesh into several tiles. PCA is performed on each tile separately. The PCA subspaces of the individual parts are then combined to a single hyperspace by adding the tile index as a dimension, such that a shape is no longer a point in a linear vector space, but a curve in the hyperspace. Shape constraints are enforced by aligning the curve of deformed shape to the closest training curve with a rigid transformation. Although the approach has been positively evaluated by the authors, some modeling choices lack rigorous mathematical justification.

Instead of building a hierarchical SSM, Koikkalainen et al. [KTL*08] propose to enlarge the training set by synthesizing artificial training examples. They devise several different methods for creating new training samples by modifying existing ones. One drawback of this approach is that the methods that synthesize training shapes use ad-hoc models without rigorous justification, and several parameters must be chosen that control these models. Also, the magnitude of additional variation must be somehow balanced with the number of additional training shapes in order to avoid the inclusion of spurious modes of deformation [CT95].

The third strategy for increasing the flexibility is to directly manipulate the shape covariance matrix. This can be seen as a regularization that accounts for the small number of training examples. Cremers et al. [CTWS02] replace non-zero eigenvalues of the singular covariance matrix with a constant which allows small shape deformations along any eigenvector of the SSM. Cootes and Taylor [CT95] use a linear finite element model (FEM) to describe additional, smooth shape deformations. Under the assumption that the parameters that control the FEM are independent and Gaussian distributed, the deformations can be easily integrated into the sample covariance matrix. Wang and Staib [WS00] propose to construct a 'smoothness' covariance matrix which they linearly combine with the training matrix. Some of these techniques have been evaluated in the study of Koikkalainen et al. [KTL*08]: In their experiments, SSMs combined with FEM performs as good as the best model that was generated with artificial training shapes. Wang and Staibs [WS00] is competitive on some, but not all organs. A reason for this might be that different landmarks may have different degrees of smoothness. This property is not represented in the model of Wang and Staib.

2.2.2.4. Variations of linear shape models

There exist several variations of the linear shape model: Several authors use orthomax rotation or sparse PCA in order to obtain sparse modes of variation $\boldsymbol{p}$ that only influence local regions [SSL06, LB07, SRR*07]. The idea behind such methods is that they allow for an easier interpretation of physicians. On the other hand, interpretability of modes is not an important issue in segmentation, whereas global shape variation is important for extrapolating the shape in local regions with low contrast.

Statistical Deformation Models [RFS03] are learned by applying PCA not on landmarks, but on control points used during nonrigid image registration. Their advantage is that they do not require to establish correspondence between landmarks. On the other hand, the control points do not necessarily coincide with boundary points and thus do not carry as much semantic information as landmarks.

Shen and Shapiro [CS09] represent shapes not by vectors, but by matrices, where each column contains the coordinates of the landmarks. A variation of PCA, 2D-PCA, is used to capture the variation of these models. They report lower reconstruction errors when using 2D-PCA compared to standard linear shape models, but do not provide an evaluation of their method in a real world application.

2.2.3. ICA-based shape models

From all machine learning approaches, Independent Component Analysis (ICA) [Com94] got arguably the most attention for non-PCA-based statistical shape modeling in the medical imaging community. It has been applied to image segmentation [UFRL03,LKK*04] and pathology detection, more particularly for the detection of myocardial diseases [SÜF*04]. The use of ICA is often motivated by the observation that ICA yields more localized modes of variation—similar to sparse PCA (see Section 2.2.2.4), but unlike PCA.

ICA is a method for *blind-source separation*: An observation vector $\boldsymbol{x}$ is explained by a vector of hidden variables $\boldsymbol{b}$. ICA assumes that the hidden variables are statistically independent from each other. One tries to determine a mixing matrix $\boldsymbol{M}$, such that $\boldsymbol{x} = \boldsymbol{M}\boldsymbol{b} + \boldsymbol{r}$, where $\boldsymbol{r}$ is a vector of noise. In contrast to PCA, the columns of $\boldsymbol{M}$ do not need to be orthogonal. To make the problem well-posed, one explicitly requires non-Gaussianity of the data, as well as unit variance for each hidden variable. ICA is more a family of techniques than a single method, because several criteria can be used to determine the mixing matrix, such as kurtosis and negentropy.

The first application of ICA to shape modeling was proposed by Bressan and Vitria [BV01]. In their work, ICA is essentially used as a technique for transforming the data into a representation on which training becomes easier. After decomposing the data into independent components, the training data is projected to each mode. For each mode, a one dimensional density is estimated using Parzen density estimation and Gaussian Mixture Modeling. The approach has only been evaluated on artificial data.

The columns of the mixing matrix $\boldsymbol{M}$ can be interpreted as modes of variation, like the eigenvectors $\boldsymbol{p}_i$ determined by PCA. In contrast to PCA, where the importance of eigenvectors is given by their corresponding eigenvalues, ICA does not provide an inherent ordering of the modes. Therefore, Üzümcü et al. [UFRL03] propose several approaches to separate ‘relevant’ from ‘irrelevant’ modes of variation. One of these techniques, which basically prefers localized modes, has been later applied in an Active Appearance Model for whole heart segmentation in cardiac MR images [UFS*03]. In their study, an ICA-based model outperforms a PCA-based model with the same number of modes of variation. In contrast to this result, Lötjönen et al. [LKK*04] report comparable segmentation accuracy of ICA and PCA-based models applied to whole heart segmentation.

2.2.4. Nonlinear shape models

Nonlinear techniques for shape modeling are usually motivated by the observation that the assumption of a Gaussian distribution does not hold for a specific training set. Although PCA-based SSMs still usually generalize very well to unseen shapes of nonlinearly distributed data sets, they can be *unspecific*: That is, it is easy to sample shapes from the model's distribution that do not represent 'plausible' instances of the modeled object class. This section reviews various approaches proposed for nonlinear shape modeling, including approaches based on polynomial regression, cluster-based approaches and methods using Kernel PCA.

2.2.4.1. Shape models based on regression and neural networks

Sozou et al. [SCTDM95] propose nonlinear shape models based on polynomial regression (PRPDM: *polynomial regression point distribution model*). The basic idea is to replace the modes of variation of linear models, which are straight lines in the space $\mathbb{R}^{dn}$, by polynomial curves. The PRPDM is computed by modifying an iterative algorithm for computing the leading eigenvectors: Instead of straight lines, polynomials curves are fitted through the residual terms which can not be represented by the current model. The training process is interactive: In each iteration, the degree of the polynomials is specified by a user. The PRPDM assumes that the second and the subsequent eigenvectors can be modeled as a function of the first [SCT*97]. Sozou et al. [SCT*97] also proposed a nonlinear SSM based on a multilayer perceptron. The network has five layers: Input and output layer, a middle layer, and two intermediate layers. The basic idea of the approach is to learn a network such that input and output are similar. Because the middle layer, called bottleneck layer, consists of very few neurons, the approach can be seen as a nonlinear dimensionality reduction technique. While the approach showed good results in the evaluation on two 2D data sets, it remains questionable whether the approach generalizes to arbitrary shapes. A clear drawback of the method is that the results depend heavily on the network structure. Moreover, as the network requires input and output neurons for each landmark vector component, the network would become very large for high-dimensional 3D shapes.

2.2.4.2. Cluster-based approaches

Several approaches for learning nonlinear shape models split the training examples into several clusters. The approaches of Bowden et al. [BMS97] and Heap and Hogg [HH97] apply standard PCA to the complete training set, cluster the training data in the principal subspace, and then apply PCA to the shapes of each cluster individually. A similar, but mathematically more rigorous approach for cluster-based shape models has been proposed by Cootes et al. [CT97]. They learn a Gaussian Mixture Model (GMM) from the training shapes using the Expectation Maximization (EM) algorithm. Because of the high dimensionality of the training data, which makes it numerically difficult to apply the EM algorithm, PCA is used for dimensionality reduction. Shape constraints are imposed by minimizing the log-likelihood of the GMM density. This approach for constraining shapes—and its disadvantages—are discussed in Section 2.4.4.4. Wang et al. [WLEH*10] combine Cootes' Mixture Modeling approach with online learning to segment the left ventricle in 4D data (cine MRI) in the presence of severe pathologies. The Gaussian Mixture Model is retrained after segmenting a frame in order to incorporate the segmentation of the frame into the model. A drawback of the method is that it requires as initialization that the first frame is already segmented, which is done manually in their work. Secondly, the authors state that the pathological, online-generated training shapes are usually far away from the original training shapes in feature space. This means that they are likely to form a self-contained cluster in the Gaussian Mixture Model. Thus, there is a risk that segmentations are only constrained by the shapes of previous frames, while the original training set is mostly ignored.

2.2.4.3. Kernel PCA

Kernel PCA [SSM98] is a nonlinear generalization of PCA. The general idea is to map vectors $\boldsymbol{x}$ with a function $\phi(\boldsymbol{x})$ to a—possibly infinite dimensional—feature space, and do a PCA of the mapped data in feature space. Since PCA can be completely formulated in terms of dot products of the training vectors, it is possible to perform a Kernel PCA without explicitly computing the mapping $\phi(\boldsymbol{x})$. Instead, a *kernel* function $k(\boldsymbol{x},\boldsymbol{y})$ is used which corresponds to the dot product in feature space, that is $k(\boldsymbol{x},\boldsymbol{y}) = \phi^T(\boldsymbol{x})\phi(\boldsymbol{x})$. In practice, one rather chooses a kernel function $k(\boldsymbol{x},\boldsymbol{y})$ which induces a particular feature space, instead of choosing a mapping $\phi(\boldsymbol{x})$ which induces a particular kernel. All methods mentioned below use a Gaussian radial basis function kernel of the form

$$k_{Gauss}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\sigma^2}\right), \tag{2.15}$$

where σ is a free hyperparameter that controls the scale of the Gaussian.

Like in PCA, where the sample mean is subtracted to get centered vectors, a centered kernel function is defined as follows:

$$\hat{k}(\boldsymbol{x},\boldsymbol{y}) = k(\boldsymbol{x},\boldsymbol{y}) - \frac{1}{s}\sum_{i=1}^{s}(k(\boldsymbol{x},\boldsymbol{x}_i) + k(\boldsymbol{y},\boldsymbol{x}_i)) + \frac{1}{s^2}\sum_{i=1}^{s}\sum_{j=1}^{s}k(\boldsymbol{x}_i,\boldsymbol{x}_j). \tag{2.16}$$

In Kernel PCA, the diagonalization $\boldsymbol{G} = \boldsymbol{\Gamma\Psi\Gamma}^T$ of the *Gram matrix* $\boldsymbol{G} = (\hat{k}(\boldsymbol{x}_i,\boldsymbol{x}_j))_{ij} \in \mathbb{R}^{s\times s}$ is computed, where $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1 \mid \dots \mid \boldsymbol{\gamma}_s)$ contains the eigenvectors and $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1,\dots,\psi_s)$ the eigenvalues of $\boldsymbol{G}$. For a vector $\boldsymbol{x}$, the kernel principal components $\boldsymbol{\beta}$ are computed by

$$\beta_i = \sum_{j=1}^{s}\gamma_{ij}k(\boldsymbol{x},\boldsymbol{x}_j). \tag{2.17}$$

Note that the complete training set is used to compute the kernel principal components of a vector $\boldsymbol{x}$.

The first Kernel PCA-based SSM has been proposed by Romdhani et al. [RGP99] who model the human face in 2D photographs from different view points. Due to the large variations introduced by the different view points, which also cause problems such as self-occlusion, linear shape models achieve unsatisfactory results. In order to constrain shapes with their nonlinear Kernel PCA shape model, Romdhani et al. [RGP99] mimick the approach that Cootes et al. [CTCG95] use for linear models: The shape is projected to its kernel principal components $\boldsymbol{\beta}$, which are constrained such that $\beta_i \in [-3\sqrt{\psi_i}, 3\sqrt{\psi_i}]$ for each i. Then, a plausible shape is generated from $\boldsymbol{\beta}$.

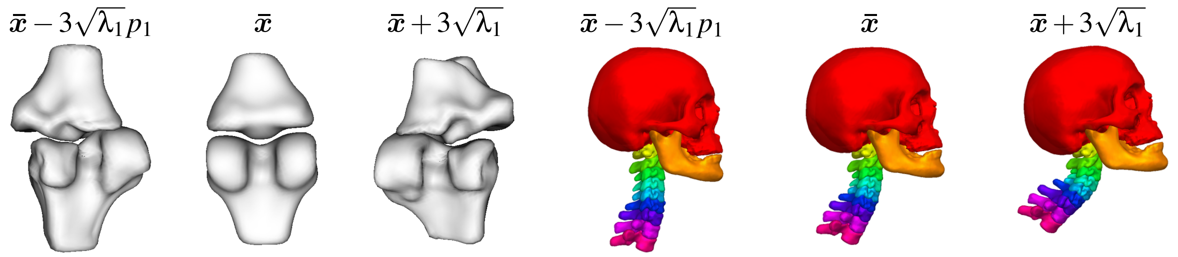
The approach of Romdhani et al. [RGP99] has two disadvantages: Firstly, there is not necessarily a corresponding point in input space for a given point in feature space, because the mapping $\phi(\boldsymbol{x})$ is in general not onto. In order to solve this so-called *pre-image* problem, a heuristic optimization technique is used which finds a shape in input shape whose kernel principal components are approximately equal to the constrained vector $\boldsymbol{\beta}$. Secondly, the β_i are approximately zero for data points far away from the training data when a Gaussian kernel is used [TT01]. This essentially means that some shapes are deemed as plausible even if they bear no similarity with the training shapes. Twining and Taylor [TT01] therefore introduce a *proximity to data* measure in the KPCA's feature space, which tends towards zero for points far away from the training data, while its local maxima lie in the vicinity of the training data. This measure is used in an optimization procedure to constrain shapes (see Section 2.4.4.4). However, the 'proximity to data' does not account for the significant loss of information that can occur when mapping a shape to the kernel principal subspace. This is especially a problem in cases where the input space has far higher dimensionality than the kernel principal subspace, whose dimension is bounded by the number of training examples. It causes that unobserved patterns that can not be expressed in the kernel principal subspace are not adequately penalized.

A third KPCA-based approach for shape modeling has been proposed by Cremers et al. [CKS03], who integrate a KPCA-based shape energy in their *diffusion snakes* framework [CTWS02] (see Section 2.5.2). Instead of using the approximated Mahalanobis Distance (Equation 2.12) as shape energy, a similar approximation of the Mahalanobis Distance in feature space is used. The approach of Cremers et al. [CKS03] avoids problems that occur in the two aforementioned KPCA-based methods [RGP99, TT01]: There is no need to solve the pre-image problem for evaluating the KPCA-based shape energy. Furthermore, the approximated Mahalanobis Distance in feature space accounts for the loss of information when mapping shapes to the kernel principal subspace.

2.2.5. Coupled and articulated shape models

Coupled Statistical Shape Models describe the joint shape variability of several objects, or several parts of the same object. For example, the geometric constraints obtained by coupling several organs can aid segmentation in cases where the contrast is very low. *Articulated models* are a special class of coupled SSMs, where different objects or parts are connected by joints.

The arguably easiest way to build a coupled SSM is to learn a linear or nonlinear SSM from joint landmark vectors, which have been constructed by concatenating the landmarks of all organs of a



(a) Joint femur and tibia shape model

(b) Articulated Atlas

Figure 2.4.: (a): First mode of variation of an SSM capturing the joint variation of femur and tibia. The mode contains a rotational component intermingled with shape deformation of the individual bones. (b): First mode of variation of an articulated atlas of the head and neck region. (Image courtesy of Sebastian Steger, Fraunhofer IGD.). While the relative pose of the bone structures changes, the shapes themselves do not deform.

patient into a single vector. Figure 2.4(a) shows the first mode of variation of a coupled model of femur and tibia learned that way. The approach has several disadvantages: Firstly, the gap between dimensionality and number of training examples becomes even larger for coupled SSMs than for single-organ models, such that there is a high risk that the model becomes too restrictive. Secondly, especially linear learning techniques are inappropriate for capturing large articulations.

To the best of my knowledge, Heap and Hogg [HH96] have proposed the first articulated shape model. They manually choose pivot landmarks and use polar coordinates for landmarks that move relatively to them. The approach assumes that articulation can be learned from the training data, which may be particularly difficult if too few training examples are available. Yokota et al. [YOT*09] therefore increase the training set of their hierarchical femur and pelvic bone model by simulating multiple poses. Each training example is copied several times, and different geometric configurations of adjacent bones are generated for each copy. For generating the copies, a model of the joint is necessary. Instead of using such a model for the construction of artificial examples and learn the articulation subsequently, the model can also be directly incorporated into the segmentation process. Kainmüller et al. [KLZH09] and Bindernagel et al. [BKS*11] follow the latter idea. Their paradigm is that “joint flexibility is not the same as shape variation” [KLZH09]. Therefore they explicitly use a physical ball-and-socket joint model for the hip [KLZH09] and knee joint [BKS*11].

Many coupled SSMs are hierarchically organized. Cerrolaza et al. [CVC12] extend the wavelet-based SSM of Davatzikos et al. [DTS03] (see Section 2.2.2.3) to decompose a joint landmark vector into signals with varying frequency. Several hierarchy levels are created, where the bottom levels describe high frequency information which is typically more localized. Most other hierarchical coupled SSMs only use two levels [RBP01, KWL*08, YOT*09]: The top level describes geometric constraints between objects, which are frequently modeled using articulated shape models. The bottom level uses single-organ SSMs that describe the variability of each object. Likewise, segmentation of these models is usually done in a two step approach: The top level model is used to obtain a rough segmentation, which is then refined with the individual bottom level models. The main difference between the various approaches is the top-level model: Yokota et al. [YOT*09] use for this a large standard SSM that contains all structures. Rok et al. [RBP01] apply PCA to the shape and pose parameters of the individual models. The problem with this is that PCA assumes that the parameters behave linearly, which is not the case for the rotation angles. Klinder et al. [KWL*08] use the spine model of Boisvert et al. [BCP*08] as top-level model: A model instance is a vector that contains relative transformations between pairs of neighboring vertebrae. One can show that these vectors build a Riemannian manifold. Riemannian statistics is applied to learn the variability of the transformations by computing Frechét mean and covariance. In contrast to the approach of Rok et al. [RBP01], Boisvert's model is based on a solid mathematical framework. A similar approach has been used by Steger et al. [SKW12] to model geometric interdependencies of bones in the head and neck region. An example of the learned atlas model is shown in Figure 2.4(b).

A different example of coupled SSMs are models that predict the shape of an organ, given the shape of others. Rao et al. [RAR08] use Canonical Correlation Analysis to derive correlations between the shape of organs. Given a segmented organ, the shape of another organ is predicted with partial least squares regression. The resulting prediction is refined using a segmentation algorithm. The approach is iterated in order to predict the shapes of other organs.

2.2.6. Discussion of shape modeling techniques

The vast majority of approaches in medical imaging use Cootes's linear shape model [CTCG95]. Although one can not expect that every training data set really behaves linearly, the linearity assumption apparently works well in many applications. The arguably biggest problem researchers observed with linear models is their limited flexibility, which is partly caused by the fact that often few training examples are available. The other reason is that linear models are usually applied in a very restrictive way. For example, the modeling technique of Moghaddam and Pentland [MP97] presented in Section 2.2.2.1, which provides a mathematically clean way to describe shape variation in a flexible way even if few examples are available, has only gained attention in classical computer vision, but not in medical imaging.

Beside the linear shape model, especially coupled and articulated shape models are subject to active research. A promising approach is here to combine Riemannian statistics with statistics in linear vector spaces in a hierarchical model: Riemannian statistics can be used to model pose dependencies between organs, and statistics in linear vector spaces to describe individual shape variation.

Up to now, nonlinear or ICA-based techniques have gained very little attention in the medical imaging community. Especially the few nonlinear approaches proposed so far have never been applied to more than small toy examples. This lack of attention can be probably accounted to the fact that linear shape modeling works quite well in practice, such that the use of more complex, nonlinear techniques do not seem worth the effort. Nevertheless, a quantitative comparison of linear and nonlinear shape modeling techniques is still missing in the literature.

2.3. Establishing correspondence

All techniques for learning shape models presented in the previous section assume that the training shapes are given as landmark vectors with inherent point correspondence. In practice, the training shapes are usually extracted from expert segmentations: For example, the Marching Cubes algorithm [LC87] can be used to convert a binary image to a densely sampled mesh. This process gives only very limited control over the number of mesh points and, more importantly, does not provide meshes with index correspondence, as each training image is converted independently. Before any of the learning algorithms presented previously can be applied, correspondence must be explicitly established on the training shapes. The problem of computing landmark vectors from a set of training shapes is known as the *correspondence problem*. Establishing correspondence means to find a representation of the training shapes such that each shape is represented by exactly n points, and such that points on different shapes with the same index correspond to each other. 'Correspond' means here that the points should represent the same anatomical feature. In this section, various different techniques that tackle the correspondence problem are presented and discussed.

In the seminal article on SSMs of Cootes et al. [CTCG95], correspondence is established by manually placing landmarks on salient points of 2D contours. Manual landmarking is very time consuming and prone to errors in case of 2D shapes—and becomes virtually infeasible in 3D. In the medical context, there is the additional problem that organs often have few salient features that can be easily identified. Reliable methods for automatic establishment of point correspondences on organ shapes are therefore indispensable for 3D shape modeling in medical imaging.

A great number of different approaches have been proposed for establishing correspondence for both 2D and 3D shapes. However, to this day, there exists no satisfying formal definition, but rather an intuitive notion of what correspondence actually is. Establishing correspondence is still an active area of research, and often new complicated data sets require the extension of existing methods or even the development of specialized methods to achieve good correspondence.

A complete review of all proposed approaches for establishing correspondence is beyond the scope of this thesis. This discussion focusses on the most popular techniques developed for 3D organ shapes, as well as concepts that have been successfully applied to 2D shapes and are readily extended to the 3D case. For further reading, the reader is referred to the recent overviews of Davies et al. [DTT08b], Heimann and Meinzer [HM09] and van Kaick et al. [vKZHCO11]. The book of Davies et al. [DTT08b] puts the emphasis on groupwise optimization. Heimann and Meinzer [HM09] especially discuss correspondence methods in the context of medical imaging. The state-of-the-art report of van Kaick et al. [vKZHCO11] discusses correspondence in the broader context of classical computer graphics.

This section is structured as follows: Section 2.3.1 introduces a common taxonomy that classifies methods into *pairwise* and *groupwise* methods. As argued therein, there are many methods that can be neither classified as purely pairwise nor purely groupwise. In this thesis, it is therefore distinguished between methods based on pairwise registration, which are discussed in Section 2.3.2, and methods based on groupwise optimization, which are explained in Section 2.3.3.

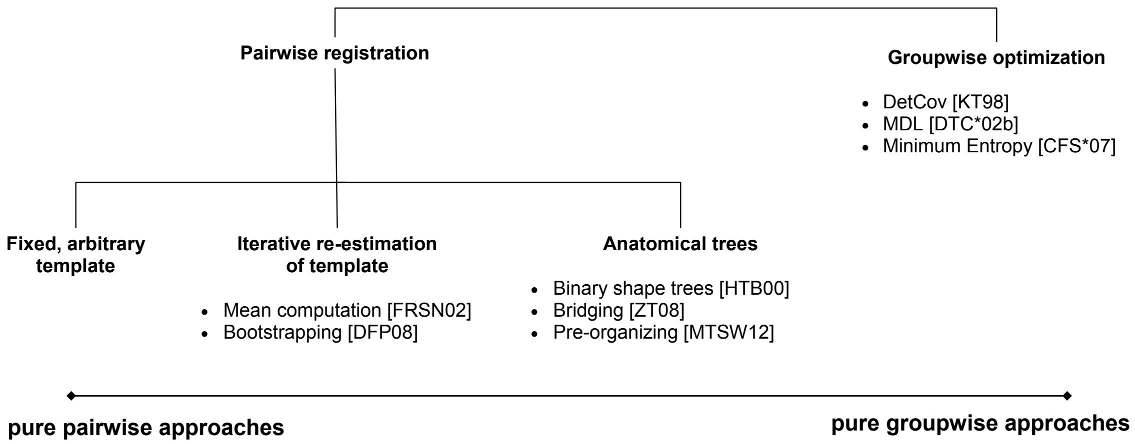


Figure 2.5.: The figure shows a taxonomy of methods for establishing correspondence which distinguishes between methods based on pairwise registration and on groupwise optimization. The registration-based methods are further divided dependent on how much they rely on group information.

2.3.1. Pairwise and groupwise approaches

The literature distinguishes between *pairwise* and *groupwise* methods for establishing correspondence (e.g., Davies et al. [DTT08b]). Pairwise methods assume that the correspondence between two shapes is independent of all other shapes in the training set. This reduces the problem to that of computing correspondences between pairs of shapes. In contrast, groupwise methods consider the training set as a whole, such that the presence or absence of a particular shape in the training set influences the correspondence between all other shapes.

Pairwise methods exploit that correspondence is transitive: If point i on shape $\mathcal{S}_1$ corresponds to a point j on shape $\mathcal{S}_2$ and point j corresponds to a point k on shape $\mathcal{S}_3$, then i and k correspond, too. Assuming this property, it is sufficient to select a single training shape, the *reference* or *template* shape, and register it to all remaining training instances in order to establish correspondence for the whole training set. In theory, the transitivity of correspondence ensures that the resulting correspondences are identical regardless of the choice of the template shape.

However, no pairwise method proposed so far is actually independent of the order in which the registration is performed, such that the quality of correspondence may vary significantly with different choices of the template. The problem is that every pairwise method is based on some kind of image or mesh registration algorithm, and none of these algorithms achieves perfect alignment. Registration algorithms are limited by the kind of transformation they compute (e.g., rigid or affine) and by the modeling assumptions that they internally use in order to approximate anatomical correspondence.

In groupwise methods, the correspondence between two shapes may also depend on other shapes in the training set. The basic idea behind groupwise methods is that every training instance provides additional information about the unknown anatomical variation that is to be modeled. In contrast to pairwise methods, this additional information is not disregarded, but exploited in order to improve the correspondence.

The term *groupwise* is mostly used in context of *groupwise optimization* approaches, which explicitly minimize a model-based objective function (see Section 2.3.3). Model-based functions evaluate correspondence based on the computed SSM, and therefore depend on every training shape. While most authors consider all methods that internally use pairwise registration per se as pairwise, many of these methods also exploit group information in order to minimize the registration error. Because the presence or absence of shapes does influence the resulting correspondences, these methods can also be regarded—at least to a certain degree—as groupwise. Figure 2.5 shows a taxonomy which orders different approaches by the degree to which they exploit group information. While this taxonomy classifies groupwise optimization methods as *pure groupwise*, it also accounts for the fact that many registration-based methods, such as those using shape similarity trees (see Section 2.3.2.4), heavily rely on information extracted from the whole training set. As the evaluation in Chapter 4 shows, exploiting shape similarity in order to determine the registration order can in fact significantly improve the correspondence quality.

2.3.2. Correspondence based on pairwise registration

The key characteristic of all methods based on pairwise registration is the use of registration for finding corresponding landmarks in pairs of shapes. Correspondence between two shapes, the *template*[3] and the *target*, is usually established in the following three steps:

1. Landmarks are defined on the template (Section 2.3.2.1).
2. The template is registered with the target (Section 2.3.2.2).
3. The landmarks are propagated from the template to the target (Section 2.3.2.3).

Because correspondence is transitive, this technique trivially induces a method for establishing correspondence for a group of $s > 2$ training shapes. An arbitrary training shape is selected as template, and its landmarks are propagated to the remaining $s-1$ shapes.

Different properties can be used to characterize pairwise registration-based methods:

Registration domain: The registration can be done using different data representations. To this end, Heimann and Meinzer [HM09] distinguish between *mesh-to-mesh*, *mesh-to-volume*, *volume-to-volume* and *parameterization-to-parameterization* registration methods.

Correspondence metric: Registration algorithms are driven by a metric that assesses the similarity of points or regions on different shapes. Thus, the metric is a key ingredient to determine correspondence.

Symmetry: The registration metric can be either symmetric or asymmetric. Asymmetric measures implicitly assume that each point on the template has one or several corresponding points on the target shape, but not the other way around. Symmetric measures take both directions into account.

---

[3] Sometimes also called *reference shape*.

Table 2.1.: Comparison of different pairwise approaches for establishing correspondence.

Method	Domain	Similarity metric	Symmetric?	Transformation	Interactive?	Parameterization?
Kelemen et al. [KSG99]	param.-to-param.	spherical harmonics	yes	reparam.	no	yes
Brett et al. [BT00]	mesh-to-mesh	closest point, normals	yes	rigid	no	yes
Lorenz et al. [LK00]	mesh-to-mesh	manual	yes	spline	yes	no
Alexa [Ale00]	param.-to-param.	manual	yes	param.	yes	yes
Andresen et al. [AN01]	mesh-to-mesh	closest point, crest lines	no	displ. field	no	no
Frangi et al. [FRSN01]	volume-to-volume	kappa metric	yes	spline	no	no
Frangi et al. [FRSN02]	volume-to-volume	label consistency	yes	spline	no	no
Lamecker et al. [LLS02]	param.-to-param.	manual	yes	param.	yes	yes
Kaus et al. [KPL*03]	mesh-to-volume	closest point, gradients	no	displ. field	no	no
Paulsen and Hilger [PH03a]	mesh-to-mesh	manual	yes	spline	yes	no
Vos et al. [VdBA*04]	mesh-to-mesh	closest point	no	rigid	no	no
Asirvatham et al. [APH05]	param.-to-param.	manual	yes	param.	yes	yes
Fripp et al. [FBM*05]	param.-to-param.	closest point	no	rigid	no	yes
Davies et al. [DTT06]	param.-to-param.	closest point	no	reparam.	no	yes
Hufnagel et al. [HEP*09]	mesh-to-mesh	prob. closest point	no	affine	no	no
Combès and Prima [CFK*11]	mesh-to-mesh	prob. closest point	yes	displ. field	no	no

Space of transformations: Registration algorithms allow different degrees of flexibility. Many algorithms restrict transformations to rigid or affine transformations. More complex, nonrigid transformation are necessary to deal with large shape variability. Nonrigid registrations can be realized using splines or regularized displacement fields. Moreover, nonrigid transformations are implicitly defined when constructing parameterization or modifying parameterizations.

Interaction: While some algorithms are fully automatic, others require different degrees of user interaction, for example to define a sparse set of corresponding landmarks.

Parameterization: Some approaches use surface parameterization. Parameterizations allow for an easy interpolation of arbitrary points on the surface. They are often used to guarantee topological consistency of the landmarks.

The choice of the correspondence metric heavily depends on the chosen registration domain. Correspondence metrics for mesh-to-mesh and parameterization-to-parameterization use mesh features, whereas the metrics employed for image-to-image and mesh-to-volume methods are based on image features. To some degree, the choice of the registration domain also influences the transformation space. Table 2.1 gives a comparison of several pairwise approaches based on the above properties.

In the following, several approaches for landmark definition, registration and landmark propagation are presented.

2.3.2.1. Landmark definition

The input shapes are in most cases very densely sampled because they are generated by automatic data acquisition processes or image-to-mesh conversion algorithms like Marching Cubes [LC87]. Landmark definition therefore often involves a downsampling process in which the same geometry is approximated with much fewer points, but without cutting of important details of the shape. For example, Frangi et al. [FRSN02] use the decimation algorithm of Schroeder et al. [SZL92] to obtain a sparse representation. Lorenz and Krahnstoever [LK00] extract a sparse template directly from a binary image. They iteratively remove surface voxels until only a sparse set remains which constitute the landmarks. The landmarks are then connected with a Delaunay triangulation. By incorporating curvature terms into the decimation criterion, high curvature features of the shapes remain preserved. If parameterizations of the input shapes are available, landmarks can be obtained by regular sampling [KSG99].

2.3.2.2. Registration

The registration of the template with the target is the key component of each pairwise correspondence algorithm. In order to ensure accurate correspondences, several authors integrate user interaction into the registration process by defining a sparse set of corresponding features on both template and target mesh [LK00, Ale00, PH03a, APH05, LLS02]. The remaining correspondences are interpolated based on this sparse correspondence relation. To this end, Lorenz and Krahnstoever [LK00] and Paulsen and Hilger [PH03a] build a thin plate spline from user-defined features which maps all landmarks of the template close to the target's surface. Other authors use surface parameterization in order to determine the remaining landmarks. A surface parameterization of a shape defines a bijective mapping of the shape to a simpler domain which allows for easy and topological consistent interpolation of points. Alexa [Ale00] first maps template and target independently to the unit sphere, and then optimizes the parameterizations such that corresponding features have approximately the same parameter space coordinates. Asirvatham et al. [APH05] propose a spherical mapping approach that directly maps corresponding features to the same parameter space coordinates. The spherical coordinates of all remaining points are determined by minimizing a stretch metric that reduces area and angle distortion of the mapping. Lamecker et al. [LLS02] use several discs to parameterize an organ surface. They identify corresponding patches by connecting the user-defined features on each shape with paths along lines with high curvature using a shortest path algorithm. Each patch is then flattened to the disc with a distortion-minimizing mapping. In contrast to spherical mapping, this approach can also handle shapes with complex, non-spherical topology, for example the pelvic bones [LSHD04].

The disadvantage of user interaction is that it is time consuming and subjective, especially on relatively smooth organ shapes that have very few characteristic feature points. Automatic methods mostly use registration algorithms in the mesh domain. Arguably the most popular and simplest algorithm for registering two shapes is the Iterative Closest Point (ICP) [BM92]. The algorithm determines a rigid transformation that minimizes a cost function based on the Euclidean distances between points on both surfaces. Many approaches for establishing correspondence directly use the ICP to define landmarks [VdBA*04] or use it as a building block in more complex approaches [BT00, DTT06, FBM*05]. Because the ICP is restricted to rigid alignments, it is usually only applicable to shape classes with moderate intra-class variability.

Various extensions of the standard ICP have been proposed for improving the correspondences. Brett and Taylor [BT00] use a symmetric ICP that does not only take distances from template to target into account, but also distances from target to template. Hufnagel et al. [HEP*09] use the Expectation Maximization-ICP (EM-ICP) [GP06] for shape model construction. The EM-ICP is embedded into a probabilistic framework and uses correspondence probabilities instead of one-to-one correspondences. They also propose an extension of the EM-ICP which estimates affine instead of rigid transformations, which allows to handle shape classes with larger variability.

For organ classes with very large shape variability, non-rigid alignment algorithms are necessary. To this end, Andresen and Nielsen [AN01] compute a smooth registration field with their *geometric constrained diffusion* algorithm. Combès and Prima [CP10] propose a nonrigid version of the EM-ICP which uses a symmetric similarity metric and estimates consistent deformation fields from template to target and back. In a later work, they show how the algorithm can be applied for establishing correspondences [CFK*11].

While most automatic registration methods compute transformation purely based on point-to-point distances, it is also possible to take additional features into account, such as surface normals [BT00] or crest lines [AN01].

Similar to interactive methods, the automatic registration can be combined with surface parameterization. For example, Davies et al. [DTT06] and Fripp et al. [FBM*05] use the ICP to compute consistent parameterizations. Kelemen et al. [KSG99] even use properties of spherical shape parameterizations in order to establish correspondence: The parameterizations of both template and target are approximated with a sum of spherical harmonics, and are aligned based on the spherical harmonics coefficients.

All approaches discussed so far register shapes in the mesh domain. An alternative approach has been proposed by Frangi et al. [FRSN01, FRSN02], who superimpose label images with non-rigid registration based on B-Splines. Kaus et al. [KPL*03] use both mesh and image for establishing correspondence. They adapt a template mesh to the expert image. Nonrigid deformation is realized by using the regularization energy of Shape Constrained Deformable Models (see Section 2.4.4).

2.3.2.3. Landmark propagation

After registration of the template with the target, landmarks must be identified on the target. In case surface parameterization is used, this process is trivial, because the landmarks on the target can be easily identified by their parameter space coordinates. For parameterization-free methods, one usually projects the landmarks of the template to the target's surface. For example, one can select the closest point to a template landmark on the target's surface as the corresponding template landmark. Such closest-point projections can lead to topological inconsistencies or undersampling of local regions of the target shape, especially if rigid alignment is used. Lorenz and Krahnstoever [LK00] therefore relax the landmarks on the target with a mass spring model that enforces that neighboring landmarks have similar distances in both target and template. Paulsen and Hilger [PH03a] model the displacement field defined by the projection of landmarks from template to target with a Markov Random Field that favors smooth displacements of neighboring points. An energy minimizing state of the field is computed with geometric constrained diffusion [AN01].

2.3.2.4. Improving pairwise correspondence using group information

Pairwise registration of a template shape to all other training shapes may introduce a bias towards the selected template. The quality of the resulting model depends highly on how representative the template is. There are two basic approaches to reduce the bias and thus to improve the model quality: Iterative re-estimation of the template and shape similarity trees. As illustrated in Figure 2.5 (see page 27), both approaches exploit group information, but shape similarity tree's rely more strongly on this additional source of information.

Iterative re-estimation of the template. In many methods (e.g. [FRSN02, HEP*09]), the template is iteratively re-estimated as follows: In the first iteration, an arbitrary template shape is chosen and registered to all other shapes. The landmarks constructed during this procedure are used to compute a mean shape, which provides the template for the next iteration. This procedure can be repeated for a fixed number of iterations, or until there are no significant changes on the mean shape after two consecutive iterations.

The bootstrapping method of Dam et al. [DFP08] is a natural extension of this averaging, which additionally exploits second order statistics: The authors propose to compute the complete shape model in the first iteration, and then adapt this deformable model to the training data in the following iterations. While this particular technique has been proposed for m-reps (see Section 2.1.2), its application to landmark-based shape models is straightforward. A possible caveat when applying such a method is that artifacts in the initial model may not disappear by simply repeating the procedure.

Shape similarity trees. The basic idea of shape similarity trees is to construct a tree that connects similar shapes. The frameworks that build upon this idea first compute a *shape similarity matrix* which assigns a distance value to every pair of shapes. The distance is computed by first registering the meshes and then computing some distance measure between the registered shapes, such as for example the Jaccard Index and/or bending energy [MTSW12]. A single comparison of a pair of shapes is usually very expensive. Because all approaches mentioned below require $O(s^2)$ comparisons between shapes, they are computationally very demanding compared to simple pairwise approaches. Once the similarity matrix has been computed, the shape similarity tree can be derived from it.

In the framework of Hill et al. [HTB00], a binary tree is constructed in a bottom-up manner. All shapes are clustered into pairs of similar shapes in a greedy manner, and each pair is averaged. Recursive execution of this clustering and averaging generates a binary shape tree which is rooted in a mean shape. Landmarks are then defined on the mean shape and propagated along the tree to the leaves. The method has been used for 3D SSM construction by Brett and Taylor [BT00].

In contrast to Hill's framework, which produces a tree with interpolated 'intermediate' shapes, Munsell et al. [MTSW12] generate a shape similarity tree which only contains the input shapes. In their *Shape Pre-organizing* framework, the tree is constructed by calculating a minimum spanning tree from the *similarity graph*, which is the weighted, complete graph induced by the shape similarity matrix. If one assumes that the distance between two shapes measures the registration error, computing the minimum spanning tree means to minimize the sum of the registration errors of all $s-1$ pairwise registrations required to establish correspondence. Shape Pre-organizing has been shown to produce far better models than just using a single template shape, even if one tries every possible shape as template shape and selects the best model. However, Dalal et al. [DSSW10] observe that the quality of the samples deteriorates with increasing edge distance to the root shape. This is because registration errors accumulate along the registration paths. Their solution to this problem is a heuristic method to compute the tree from the similarity graph, which tries to retain links between similar shapes while at the same time keeping the path lengths small.

The *bridging* method of Zha and Teoh [ZT08] is a special case of Munsell's framework, where the path length from the root shape is at most two. In cases where the template and a training shape are dissimilar, landmarks are indirectly propagated by using a third shape as a bridge. The requirement is to choose the bridge in such a way that it is both similar to the template and the training shape.

2.3.3. Groupwise optimization

While the methods discussed in Section 2.3.2.4 use a pairwise registration algorithm as a building block, groupwise optimization methods follow a different approach. The idea is to explicitly optimize an objective function that depends on all training shapes. Groupwise optimization has been pioneered by the work of Kotcheff and Taylor [KT98] who have devised the first *model-based* objective function, which measures correspondence based on properties of the SSM computed from the training set $\mathcal{X}$. Assuming a Gaussian shape distribution (Section 2.2.2), Kotcheff and Taylor measure the correspondence by the function

$$F_{\text{DetCov}}(\mathcal{X}) = \log\left(\prod_{i=1}^{s-1}(\lambda_i+\varepsilon)\right) = \sum_{i=1}^{s-1}\log(\lambda_i+\varepsilon). \tag{2.18}$$

The name *DetCov* alludes to the fact that the determinant of the covariance matrix $\det(\boldsymbol{\Sigma})$ is equal to the product of its eigenvalues. Because the covariance matrix is usually singular in practice, that is $\det(\Sigma) = 0$, the computation of DetCov is restricted to non-zero eigenvalues of $\boldsymbol{\Sigma}$. The regularization parameter $\varepsilon > 0$ is included for the case that an eigenvalue approaches zero during optimization.

The DetCov function is motivated by *Occam's Razor*, which states that from two different hypotheses that explain an observation equally well, one should prefer the simpler one. DetCov implements this principle by minimizing the 'volume' that the samples occupy in high dimensional space, measured by the product of variances (eigenvalues) on the principal axes. Thus, DetCov favors compact models. However, no rigorous justification of the DetCov function is known [DTC*02b].

Davies et al. [DTC*02b] therefore derive a model-based objective function based on information theory by applying the *Minimum Description Length* (MDL) principle, which can be seen as a formal implementation of Occam's Razor. MDL measures model complexity by the coding length of a model, favoring models with small coding length. Like the DetCov function, Davies's MDL function is based on the assumption that the shapes are multivariate Gaussian distributed. Additionally, it requires a quantification parameter which specifies with which accuracy continuous data is approximated, such that one can compute how many bits are required to transmit both the model parameters and the training data. A detailed derivation of the MDL function can be found in the book of Davies et al. [DTT08b]. Davies et al. [DTCT10] also propose the following approximation[4] of the full MDL function, which is used in the experiments in Chapter 4:

$$F_{\text{MDL}}(\mathcal{X};\Delta) = \sum_{i:\lambda_i \geq \lambda_{\min}} (s-2)\log(\sqrt{\lambda_i}) + \sum_{i:\lambda_i < \lambda_{\min}} \left[(s-2)\log(\sqrt{\lambda_{\min}}) + \frac{s\lambda_i}{2\lambda_{\min}} \right]. \tag{2.19}$$

Here, Δ is the coding accuracy of the data, and $\lambda_{\min} = 4\Delta^2$. In order to remove the influence of a specific coding accuracy, one integrates over an interval of coding accuracies:

$$F_{\text{MDL}}(\mathcal{X};\Delta_{\min},\Delta_{\max}) = \int_{\Delta_{\min}}^{\Delta_{\max}} F_{\text{MDL}}(\mathcal{X};\Delta)\mathrm{d}\Delta. \tag{2.20}$$

The interval $[\Delta_{\min},\Delta_{\max}]$ can, for example, be chosen in dependence of the voxel resolution of the image data from which the training shapes have been extracted.

The MDL objective function is quite similar to DetCov, and one can regard DetCov as an approximation to MDL. An alternative approximation,

$$F_{\text{Thodberg}}(\mathcal{X}) = \sum_{i:\lambda_i \geq \lambda_{\min}} \left(\log\left(\frac{\lambda_i}{\lambda_{\min}}\right) + 1 \right) + \sum_{i:\lambda_i > \lambda_{\min}} \frac{\lambda_i}{\lambda_{\min}} \tag{2.21}$$

was proposed by Thodberg [Tho03].

Both Kotcheff and Taylor [KT98] and Davies et al. [DTC*02b] assume a continuous shape surface such that their objective function is independent of the number of landmarks used in the shape model. The covariance matrix is computed by integrating over the shapes' surfaces. Note that even in the case of continuous shape surfaces (infinitely many landmarks), the number of non-zero eigenvalues remains bounded by the number of training examples, and can be computed using Dual PCA (see Section 2.2.2). In theory, the assumption of a continuous shape surface removes the issue of representation accuracy. If finitely many landmarks are used, it must be ensured that these landmarks preserve all details of the shapes well. Because the model-based objective functions presented so far are designed

---

[4] For simplicity, constant terms of the original formula are omitted in Equation 2.19, because they have no influence on the minimization result.

to obtain simple models, they tend to favor shapes with less details, such that certain regions of the shapes may become undersampled in course of optimization. Thodberg adds a ‘stabilizing term’ to the MDL function, but does not provide any formal justification for it. A more rigorous approach is the entropy minimization of Cates et al. [CFS*07]. They propose to minimize the entropy of the distribution of a group of shapes, while at the same time maximizing the accuracy of the individual surface representations

$$F_{\text{Entropy}} = \sum_{i=1}^{s-1} \log(\lambda_i + \varepsilon) - \sum_{i=1}^{s} \sum_{j=1}^{n} \log\left(\frac{1}{n(n-1)} \sum_{k \neq j} G(w_j(\boldsymbol{x}_i^j - \boldsymbol{x}_i^k), \sigma_j) \right). \tag{2.22}$$

The first sum measures the entropy of the group, and is identical to the DetCov function. The second sum accounts for the representation accuracy, which is also expressed in terms of entropy minimization: The landmarks $\boldsymbol{x}_i^j$ are considered to be a set of particles which repulse each other but are constrained to lie on the shape’s surface. The distribution of the particles is modeled using Parzen Windowing, where $G(w_j(\boldsymbol{x}_i^j - \boldsymbol{x}_i^k), \sigma_j)$ is an isotropic Gaussian with scale σ_j. The weights w_j control how densely particular regions of the shapes are sampled: Uniform weights result in a uniform shape sampling, while weights depending on local curvature allow for a curvature adaptive shape sampling.

Groupwise objective functions do not necessarily have to depend on the spatial variability of the shapes, but also on other features, such as local curvature [SOHG08] or application specific information, such as connectivity information derived from Diffusion Weighted Imaging [ONC*09]. Groupwise objective functions such as MDL can also be generalized to non-Gaussian shape distributions, for example those based on Kernel PCA [CZS10].

2.3.3.1. Optimization by reparameterization

Groupwise objective functions such as MDL are usually minimized using the *optimization by reparameterization* paradigm pioneered by Kotcheff and Taylor [KT98]. Figure 2.6 illustrates this principle: All shapes are mapped to a common parameter domain, for example the unit sphere. For each shape, sampling points are defined in order to sample landmark vectors from the parameter domain. With these landmark vectors, the objective function can be calculated. By *reparameterization*, that is by modification of the parameterizations, the objective function is minimized. Most other authors adopted the principles of this approach. An alternative optimization method was proposed by Cates et al. [CFS*07] who minimize Equation 2.22 by varying the positions of a set of particles on the shapes’ surfaces.

For 3D shapes, Davies et al. [DTC*02a] propose an iterative stochastic optimization scheme. In each iteration, hyperparameters that control the reparameterization functions are chosen stochastically. Then the objective function is minimized by reparameterizing the sampling points of one or more shapes, for example using the Nelder-Mead simplex algorithm [DTC*02a] or gradient descent [HWWM05]. Gradients of the simplified objective function such as $F_{\text{DetCov}}(\mathcal{X})$ or $F_{\text{Thodberg}}(\mathcal{X})$ can be computed analytically, while the gradient of the reparameterization method is estimated numerically. The iterative scheme terminates after a fixed number of iterations or if convergence is reached. In order to achieve faster or more robust convergence, multiresolution optimization schemes can be employed in

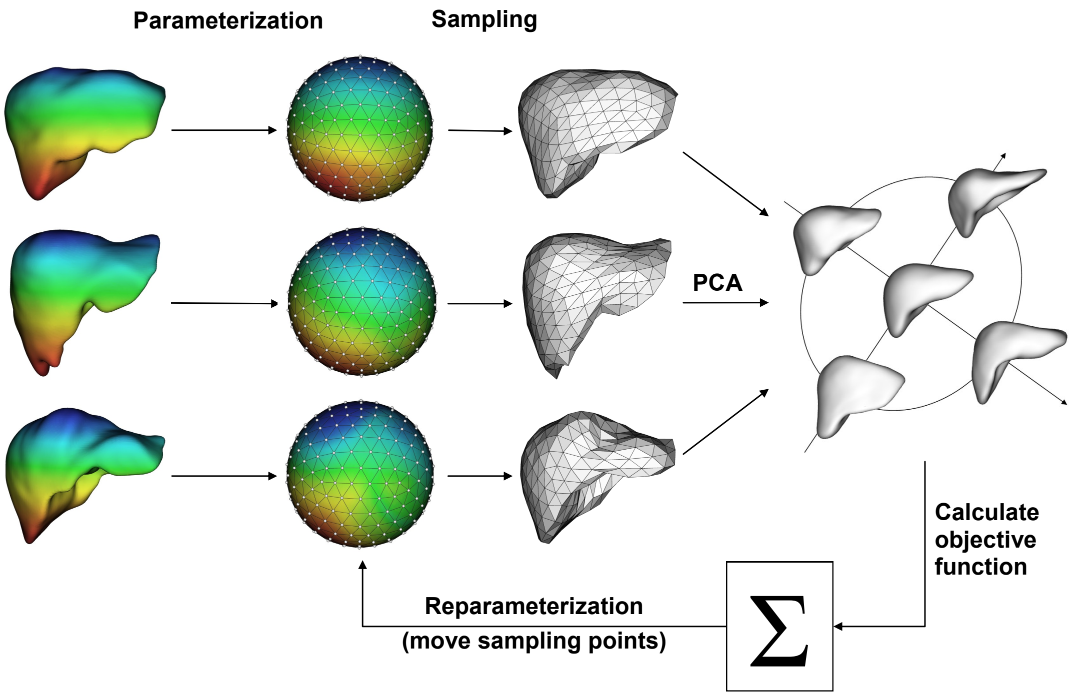


Figure 2.6.: The groupwise optimization pipeline. Each shape is mapped to a parameter space, for example the unit sphere. The objective function is optimized by iterative modification of the sampling points.

which the reparameterizations become more localized in later stages of the optimization. A detailed analysis of different optimization strategies and their influence on the optimization has been made by Davies et al. [DTCT10].

2.3.3.2. Shape parameterization and reparameterization

For optimization by reparameterization, a suitable parameter domain for the training shapes has to be chosen. The choice depends for 3D organ shapes on their *genus*, which intuitively corresponds to the number of holes in the surface. Since many organs, such as liver, heart or kidney, have no holes and are therefore of genus 0, virtually all approaches for groupwise optimization in 3D use the unit sphere as a parameter space. In their book, Davies et al. [DTT08b] show conceptually how MDL optimization can be applied to open surfaces, that is, surfaces with a boundary.

The mapping of a shape to the unit sphere, or other parameter domains, usually introduces distortion of angles and areas. While it can be shown that there always exists a conformal (angle preserving) mapping of a shape with genus 0 topology to the unit sphere, there exists in general no isometric mapping, that is, a mapping that preserves both angles and areas [HLS07]. Davies et al. [DTCT10] compute spherical parameterizations with the quasi-isometric method of Brechbuehler et al. [BGK95]. In this constrained optimization approach, area preservation is enforced by explicit constraints, while the objective function penalizes angular distortion. Other authors use conformal mapping instead [HWWM05, FBM*05]. The drawback of conformal mapping is that it can result in large area distortions, which makes sampling of landmark vectors from the parameter domain difficult. For the spherical mapping technique of Brechbuehler et al. [BGK95], it suffices to sample the unit sphere uniformly in order to obtain landmark vectors that preserve the geometry of the input shapes well. An approximately uniform sampling of the unit sphere can be generated, for example, by subdividing the unit icosahedron and projecting its point to the sphere [DTT08b]. In contrast, uniform sampling on conformal parameterizations leads in most cases to poorly sampled landmark vectors. This can be partially alleviated by using an adaptive sampling scheme based on area distortion maps [HWM07].

A second issue is that the mapping to the sphere should be *consistent*: When parameterizing a group of shapes, the same anatomical regions should be mapped to approximately the same spherical regions. Davies et al. [DTT06] report that inconsistent initial parameterizations increase the computation time and may even cause the optimization to converge to a poor local optimum, such that the resulting model is implausible. They propose the following method for generating consistent spherical parameterizations: They first compute an area-preserving parameterization of a template shape. All other shapes are aligned to the template with the ICP. Using the closest point relation derived from the ICP algorithm they define an objective function which penalizes the squared Euclidean distance of spherical points of a shape to corresponding spherical points of the template. This objective function is optimized by reparameterization. Fripp et al. [FBM*05] also propose to use ICP for aligning spherical parameterizations in context of groupwise optimization. In their approach, all shapes are first parameterized independently. The ICP-based correspondence relation is used to compute rotation matrices which are used to align the parameterizations.

During optimization, the shape parameterizations have to be modified using a reparameterization function, which can be any diffeomorphic function from the parameter domain to itself. A diffeomorphic functions is an invertible function such that both the function and its inverse are differentiable. This property ensures that the reparameterization is smooth and preserves the topology. To this end, most authors propose parametric functions such as *Theta reparameterization* [DTC*02a], *Gauss Warps* [HWM07], and *Clamped Plate Splines* [DTT06]. Davies et al. [DTT08a] have also devised a non-parametric approach, which allows for a broader class of reparameterizations than parametric functions.

2.3.3.3. Pitfalls of groupwise optimization

As discussed above, the issue of representation accuracy can be neglected in theory if the shapes are modeled as continuous surfaces, because the objective function is computed by integration over the complete surface. In practice, the integral must be approximated numerically, and here the representation issue recurs. Because model-based objective functions tend to simplify the geometry of the shapes, sampling points tend to collapse in some regions of the shape, while other regions become

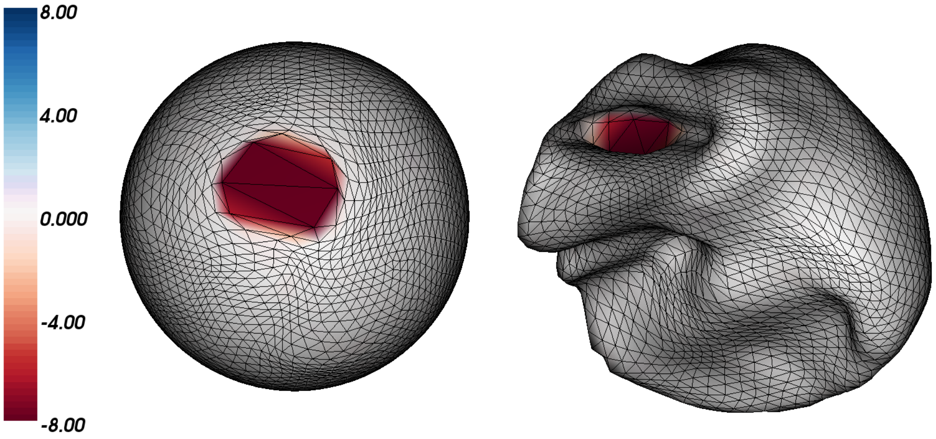


Figure 2.7.: Reparameterization during groupwise optimization often introduces large distortions, as shown in the sampling pattern on the left, which was uniform prior to optimization. The corresponding landmark vector shown on the right undersamples the corresponding high curvature region of the original shape. The color mapping illustrates the area distortion, measured by $A_M(t) - A_P(t)$, where $A_M(t)$ and $A_P(t)$ are the normalized areas of triangle t on mesh and sampling pattern, respectively. The pattern has been computed with the optimization algorithm described in Section 3.2.4.

undersampled [DTCT10]. Figure 2.7 shows a sampling pattern with a highly distorted region and the corresponding landmark vector sampled with this pattern from the parameter space. Such distortions typically occur in high curvature regions of the shape, because the objective function tries to simplify the model and therefore moves the points to less complex regions.

Davies et al. [DTCT10] propose a numerical integration scheme based on a finite number of sampling points, where an area measure on the mean shape is used to account for sampling problems. The basic idea is to weight a sampling point proportional to the area of its neighboring triangles on the mean shape. However, the algorithm that created the sampling pattern shown in Figure 2.7 follows this strategy, which means that it does not guarantee to avoid sampling problems. A complementary approach has been proposed by Heimann et al. [HWM07] who redistribute the sampling points every time after a fixed number of iterations. In their approach, they create area distortion maps for every parameterization, average these maps, and adaptively distribute the new sampling points according to the averaged distortion map. The problem here is that the redistribution is relatively time-consuming, and that some distortions may not be adequately handled due to averaging. Yet another approach to avoid landmark collapsing is to keep the parameterization of one *template* example fixed [DTC*02b], but as Thodberg [Tho03] notes, the significance of the fixed template diminishes with increasing number of training shapes.

2.3.4. Evaluation of model quality

The large number of different solutions for the correspondence problem raises the demand for methods that quantitatively evaluate the quality of SSMs learned from the same data, but with different correspondences. To this end, Davies [Dav02] has established the measures *compactness*, *specificity* and *generalization ability*. For their computation in 3D, it is assumed that the following data is available:

1. A set of shapes, represented as triangle meshes $\{M_1, \dots, M_n\}$,
2. a set $\mathcal{X}$ of landmark vectors corresponding to these shapes and
3. a linear model computed from $\mathcal{X}$, specified by the mean shape $\bar{\boldsymbol{x}} \in \mathbb{R}^{3n}$, the eigenvector matrix $\boldsymbol{P} \in \mathbb{R}^{3n \times t}$ and the corresponding eigenvalues $\lambda_1, \dots, \lambda_t$.

Moreover, let $\langle M, M' \rangle$ denote a distance metric for comparing two shapes.

Compactness: Compactness measures the complexity of the model. Small values correspond to less complex models. Compactness is measured by the sum of the model's eigenvalues:

$$\mathsf{C} = \sum_{i=1}^{t} \lambda_i. \tag{2.23}$$

Specificity: A shape model is *specific* if it describes only shapes of the modeled object class. In a probabilistic model, shapes of the object class should have a higher probability than other shapes. Specificity of a linear model is computed by sampling l shapes from the Gaussian probability distribution of the linear shape model and comparing them to all training examples. A low error is obtained if each drawn sample is similar to at least one training shape:

$$\mathsf{S} = \sum_{i=1}^{l} \min_{\{j \in 1, \dots, s\}} \langle M(\boldsymbol{y}_i), M_j \rangle . \tag{2.24}$$

Here, the $\{\boldsymbol{y}_i\}$ are l landmark vectors sampled from the probability distribution of the SSM. A theoretical basis for specificity, including a discussion of higher order neighborhoods, different distance functions and links to density estimation, is given by Twining and Taylor [TT11].

Generalization ability: The generalization ability measures how good an SSM generalizes to unseen shapes. All shapes of the modeled object class should have a high probability, not only the training examples. There exist two different approaches for measuring the generalization ability of an SSM. One approach is similar to computing specificity, but changes the roles of samples and training shape:

$$\mathsf{G} = \sum_{i=1}^{s} \min_{\{j \in 1, \dots, l\}} \langle M_i, M(\boldsymbol{y}_j) \rangle . \tag{2.25}$$

Alternatively, the generalization ability can be computed using leave-one-out tests. Let $\bar{\boldsymbol{x}}_{-i}$ be the mean shape and $\boldsymbol{P}_{-i}$ the eigenvector matrix of the SSM learned from all but the i-th sample, and let $\boldsymbol{x}'_i = \bar{\boldsymbol{x}}_{-i} + \boldsymbol{P}_{-i}\boldsymbol{P}^T_{-i}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_{-i})$. The leave-one-out generalization ability is defined as

$$\mathsf{G}_{\text{loo}} = \sum_{i=1}^{s} \left\langle M_i, M(\boldsymbol{x}'_i) \right\rangle . \tag{2.26}$$

It is common practice to compute compactness, specificity and generalization ability several times for a given set of landmark vectors $\mathcal{X}$, each time using a different number of modes of variation. For example, if the model was trained from s examples, the measures are computed for $t = 1, \ldots, s-1$.

Most quantitative evaluations of correspondence algorithms use specificity, generalization ability and compactness nowadays. However, Ericsson and Karlsson [EK07] and Munsell et al. [MDW08] report examples with counter-intuitive evaluation results when using these measures. They propose alternative correspondence measures which are based on the assumption that ground truth data is available. However, obtaining ground truth correspondences for 3D shapes from the biological or medical domain is virtually impossible and highly impractical.

The observations made by Ericsson and Karlsson [EK07] and Munsell et al. [MDW08] are based on the assumption that shapes are compared with the Euclidean norm $\|\boldsymbol{x} - \boldsymbol{x}'\|^2$, as it has been originally proposed by Davies [Dav02]. In fact, using the Euclidean norm has some disadvantages: It is dependent on the number of landmarks, and, more severely, does not take differences between the original input meshes M_i and the corresponding landmark mesh $M(\boldsymbol{x}_i)$ into account. If $M(\boldsymbol{x}_i)$ undersamples certain regions of M_i and thus lacks some details, the model tends to get less complex and therefore often reaches better metrics than models that preserve all geometric details well.

These problems can be addressed by using other measures for shape comparison. To this end, any volumetric or surface based measure for assessing segmentation quality can be used for this purpose. Such measures are discussed later in Section 2.6. In context of correspondence evaluation, Heimann et al. [HWM07] use the Jaccard Index [Jac01] to compare shapes. Because the Jaccard index is a volumetric measure, the shapes must be converted into voxel grids for computation. Gollmer and Buzug [GB10] use surface based measures, which are computed by Monte Carlo integration over the shapes' surfaces.

It should be noted that as long as SSMs are described in terms of a probability distribution from which shapes can be sampled, it is possible to use the measures generalization ability and specificity to compare linear and nonlinear SSMs. However, this does not include nonlinear models based on KPCA, because their density is defined in feature space, not in the input space $\mathbb{R}^{dn}$.

2.3.5. Discussion of algorithms for establishing shape correspondence

The complexity of 3D shapes makes the use of (semi-)automatic approaches for establishing correspondence indispensable. Interaction is only sensible if the user only needs to specify a few salient points. Such interactive methods help to make the registration more robust and can thus improve the correspondences. Their drawback is that consistent specification of only a few landmarks on a set of training shapes is very time consuming and only reproducible if salient features can be easily identified. Because of that, an automatic approach to the correspondence problem has been chosen in this thesis.

As discussed above, methods for establishing correspondence can be classified into two major categories: Those based on pairwise registration, and those using groupwise optimization. The appeal of groupwise optimization techniques is that they are built on a solid mathematical foundation and make the correspondence measure explicit through an objective function. Moreover, the few evaluation studies that compare 3D groupwise optimization with registration-based methods suggest that groupwise methods achieve a higher model quality. Styner et al. [SRN*03] report that optimization using DetCov or MDL achieves a better model quality than SPHARM as well as an interactive subdivision technique. Similarly, in a recent application driven evaluation of Gollmer and Buzug [GB12], a higher segmentation accuracy is achieved when using an optimized model. On the downside, groupwise optimization is extremely time-consuming: It can take days or several weeks to optimize a 3D SSM. Moreover, the fitness landscapes of groupwise objective functions are not well understood, and many researchers observe that there is a high risk that the optimization gets stuck in poor local minima [DTT06, KGWB11].

A clear advantage of registration-based methods is that they are much faster. Even nonrigid methods usually only require not more than a few minutes for registration, such that an SSM of reasonable size can often be constructed within an hour. Using techniques such as shape similarity trees, group information can be integrated into the model building process in order to avoid large registration errors by only registering similar shapes. A quantitative comparison of shape similarity trees with groupwise optimization is provided in Chapter 4. To the best of my knowledge, this is the first time that these techniques are compared.

Finally, it should be noted that groupwise optimization by reparameterization requires that all shapes are mapped to a common parameter domain. The consistent parameterization methods designed for this purpose are based on pairwise registration. This means that improving pairwise correspondence methods is also worthwhile when the ultimate goal is to construct models using groupwise optimization: By using better pairwise methods for initialization, the optimization time is decreased and convergence to poor local optima is avoided.

2.4. Image segmentation with Active Shape Models

One of the most important applications of SSMs is image segmentation, where the SSM is used to constrain segmentations to the class of shapes learned from the training examples. This section presents the *Active Shape Model* (ASM), a segmentation algorithm which iteratively adapts a shape model to an image. The ASM is the most frequently used SSM-based algorithm in medical imaging. Alternative segmentation algorithms that also incorporate shape priors are discussed in Section 2.5.

2.4.1. Overview

The ASM [CTCG95] is an iterative algorithm for image segmentation, which employs two different kinds of models: A shape model, which encodes prior knowledge about the object's shape, and an appearance model, which describes the appearance of the object's boundary in images. The latter model is necessary to adapt the model mesh to the image.

Algorithm 1 formulates the ASM in a generic way. For each step, there exist various alternative approaches, which are discussed in the sections below. The algorithm starts with placing an estimate of

Algorithm 1 The Active Shape Model

```
1: Place a valid instance of the shape model into the image
2: while not converged do
3:     for all landmarks do
4:         Use appearance model to detect an optimal image feature in the landmark's vicinity
5:         Displace landmark to detected image feature
6:     end for
7:     Constrain deformed shape with shape model
8: end while
9: Postprocess the result (optional)
```

the target's shape—usually the SSM's mean shape—into the image (Section 2.4.2). In each iteration of the main loop, the appearance model is used to perform a local search in the vicinity of each landmark in order to search for optimal image features (Section 2.4.3). The shape is then deformed by displacing the landmarks to the detected image feature. Since this deformed shape does not necessarily correspond to a plausible object instance, the deformed shape is constrained with the SSM (Section 2.4.4).

The ASM can be regarded as a variation of the Expectation Maximization algorithm [CT04]. From this point of view, the search for image features and the displacement of the landmarks corresponds to a (heuristic) E-step (line 2-6), while the M-Step updates the model hypothesis, that is, the shape prior (line 7).

2.4.2. Model initialization

The ASM is a local search algorithm with limited capture range. For robust and accurate segmentation, it is necessary to determine an initial position of the model in the image, that is, the model must be placed roughly onto the target structure. Model initialization can be achieved with user interaction, for example by letting the user define various points [KSG99]. For automatic model initialization, many authors exploit domain knowledge or additional application specific information. For example, Kainmüller et al. [KLL07] initialize their liver segmentation algorithm for CT data by identifying the right lung lobe by voxel counting. The lungs lobes can be relatively easily identified in CT data because they correspond to very large regions with low intensity in the images. Toth et al. [TTR*11] initialize their prostate segmentation algorithm for T2-weighted MRI scans by exploiting additional information from a second modality, Magnetic Resonance Spectroscopy (MRS). Although the spatial resolution of the MRS data is much lower than that of MRI scans, it is sufficient for identifying a region of interest. Moreover, it is easier to discriminate between prostate and non-prostate tissue based on MRS spectra than based on MRI intensities. The fundamental disadvantage of such application-driven approaches is that they do not generalize to other structures and applications. Moreover, they usually rely on hand-selected and manually tuned parameters, such that there is a higher risk that they fail on unseen data.

A more general solution to the problem of object detection is to use evolutionary algorithms, as proposed by Heimann et al. [HMMW07]. They evolve a population of hypotheses, each defined by pose and shape parameters. Mutation of a hypothesis is achieved by adding random Gaussian noise to the vectors. The fitness of a hypothesis, which determines its chances to reproduce, is computed based on the appearance models of the ASM. The evolutionary initialization has been successfully applied to liver segmentation in CT images [HMMW07] and to prostate segmentation in transrectal ultrasound [HBS*08]. In the latter application, the segmentation fails in some cases, which suggests that the initialization method lacks robustness. An additional disadvantage is that evolutionary initialization takes several minutes, which is quite slow when compared to techniques like Marginal Space Learning which is discussed below.

Another approach to automatic model initialization is the use of the Generalized Hough Transform (GHT) [Bal81], which is a method for detecting an arbitrary template shape in an image. Because a single template is often insufficient for detecting an organ with considerable shape variation, Brejl and Sonka [BS00] use the complete training set as templates to build the detector. Ecabert et al. [EPS*08] propose an improved version of Brejl and Sonka's shape-variant GHT for whole heart localization in CT scans. They also exploit additional knowledge about the patient's pose to restrict the search space and thus get faster detection times, such that the heart is detected within two seconds on a standard desktop PC. Ruppertshofen et al. [RCS*11] propose the Discriminative Hough Transform, in which boundary points of the shapes have different weights. They use an iterative training procedure that alternates between detector construction and detection. By incorporating misdetections into the next detector construction, the weights are adapted such that the most discriminative boundary points get a stronger impact.

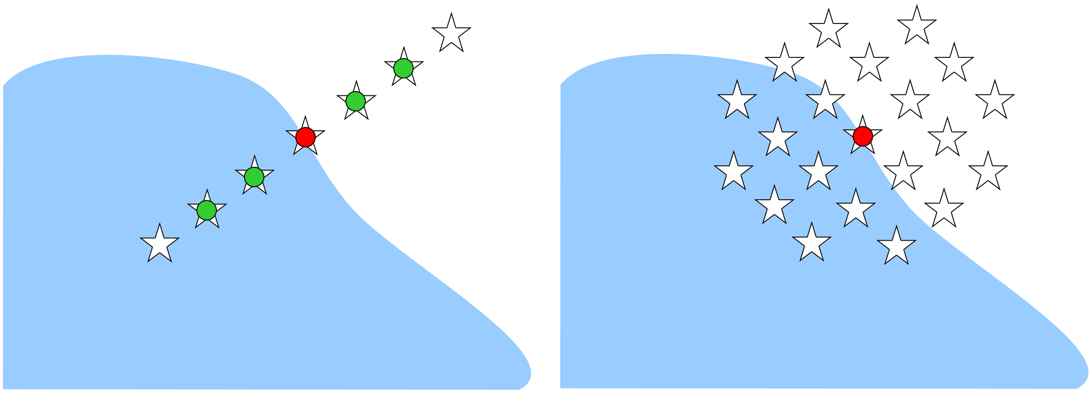
There is also an ongoing trend to translate machine learning-based object detection algorithms from classical computer vision to medical imaging. The face detection algorithm from Viola and Jones [VJ01] builds a basis for many approaches. The Viola-Jones algorithm slides a detector window over the complete image and decides for each subimage under the current detector position whether it contains a face or not. For training the classifier, AdaBoost is used to select the most discriminative features from an overcomplete set of 2D-Haar-like features. By arranging the classifier in a cascade, the detector is very fast, such that real-time face detection becomes possible. The problem of organ detection in CT images is very similar to that of face detection: The pose of an organ has usually not much variation due to the CT scanning protocol. This corresponds to the assumption in face detection that faces in natural images are mostly upright. Moreover, the image intensities in CT are standardized through the Hounsfield scale, which is a great advantage when computing Haar-like features, because no intensity normalization is necessary. Wimmer et al. [WSH09] use the Viola-Jones algorithm in order to initialize their level set algorithm for liver segmentation. Although Wimmer's algorithm works on tomographic scans, detection is done per slice: The slice with the largest detected liver region is used for initialization. Tu et al. [TZB*06] use 3D-Haar-like features for polyp detection and learn a probabilistic boosting tree [Tu05] instead of a cascade. Because polyps can be arbitrarily rotated in CT scans, they additionally propose a sample alignment in order to remove the rotational component.

For both localization and complete pose estimation of organs, Zheng et al. [ZBG*08] proposed Marginal Space Learning (MSL). They model the pose of an object in 3D by $(t_x, t_y, t_z, \psi, \phi, \theta, s_x, s_y, s_z)$, where (t_x, t_y, t_z) is a translation vector, ψ, ϕ, θ are rotation angles, and s_x, s_y, s_z are factors for anisotropic scaling. The basic idea of MSL is to estimate the parameters step by step instead of estimating them jointly, that is, first the translation vector is estimated, then the rotation parameters, and finally the scale factors. For both scale and rotation estimation, only the best hypotheses of the previous step are investigated, such that detection can be done very quickly. The object detector that estimates the translation is a probabilistic boosting tree which is trained with 3D Haar-like features. As Haar-like features can not be efficiently evaluated when rotating the volume, estimators for rotation and scale parameters are trained with so-called steerable features, which are computed by rotating and scaling a fixed sampling pattern. MSL has been successfully applied to the detection of various organs, including the heart [ZBG*08], and the liver [LZZ*08]. Additionally, improvements to the original MSL have been proposed that use quaternions for a more robust orientation estimation and that exploit further constraints in order to achieve faster detection results [ZGL*09]. Organ detection in 3D volumes with MSL requires only a few seconds on modern desktop computers.

Instead of using Haar-like features, Erdt [Erd12] introduces Histograms of Oriented Gradients and Intensities (HOGIs) as features. These features are based on the popular Histograms of Oriented Gradients (HOGs) [DT05], which have been successfully applied to pedestrian detection in natural images. For computing HOGs, the detection window is subdivided into several subwindows. For each subwindow, a histogram is created by computing oriented gradients for each pixel in the subwindow and discretizing them. The final HOG descriptor is the concatenation of the normalized histograms of all subwindows. Erdt additionally enriches HOGs with intensity histograms. This is advantageous because CT imaging provides—in contrast to natural images—a normalized intensity scale. Instead of using a linear SVM for classification as it is done in pedestrian detection [DT05], Erdt [Erd12] uses boosted Classification and Regression Trees. Compared to Haar-like features, the computation of HOGIs are computationally more expensive during detection. The reason for this is that only few Haar-like features have to be computed for detection, while for computing HOGIs, the whole subwindow must be taken into account. On the other hand, feature selection is mandatory when using Haar-like features, which makes the training phase more complex.

The performance of a single object detector can be further improved by coupling object detectors of different organs. Seifert et al. [SBZ*09] use a Markov Random Field to encode geometrical relationships between several organs. These relationships are imposed as additional constraints on the object detectors such that the detection results are anatomically consistent. Erdt [Erd12] uses PCA to learn relative positions of several organs. Starting with several detected organ positions for each structure, outliers that do not adhere to the learned constraints are iteratively discarded, until only one position for each structure remains.

Although one can not expect perfect detection rates from any of the aforementioned automatic methods, object detection in medical images has pretty much matured in the last few years. Methods based on machine learning such as MSL stand out from other methods because of their robust and fast detection rates and their high versatility. A challenge for object detection are pathologies in the images. For



(a) Linear search (b) Ball-shaped search

Figure 2.8.: (a): Linear feature search around a landmark (red circle). The stars are candidates on which image features are sampled. An image feature is typically an oriented 1D profile. The figure shows a linear profile consisting of five sampling points (circles) which is centered at the landmark. (b): Ball-shaped feature search.

example, Ecabert et al. [EPS*08] report a single misdetection on 150 test scans of their GHT, which they attribute to a large aneurysm in the data set. Combining several organ detectors with geometric constraints therefore seems to be a reasonable way for increasing the robustness, even if it suffices for the given application to segment only a single organ.

2.4.3. Image feature search

After having initially placed the model mesh into the image, it is iteratively adapted to the image by alternating between image feature search and enforcement of shape constraints. Image features are sampled in the vicinity of each landmark at different positions, which are called *candidates* in the following. An image feature can be a scalar, such as the image intensity or the gradient magnitude of a voxel, or a high-dimensional vector. Figure 2.8(a) shows a one-dimensional profile that is centered at the feature position and oriented along the surface normal. Such profiles are used in most implementations to sample vector-valued image features.

The search for optimal image features can be divided into three steps: Firstly, the search strategy defines how many candidates are sampled, and where they are sampled. Secondly, the fitness of each candidate is assessed by an appearance model. Finally, one candidate is chosen for each landmark during image feature selection.

2.4.3.1. Search strategies

The standard search strategy is to sample several image features on a line that is centered at a landmark and oriented along the landmark's normal, as illustrated in Figure 2.8(a). Free parameters of this linear search strategy are the number of sampling points on the line and the distance between the points. Several authors refine the linear search by rule based systems, thus excluding search directions or stopping the search when a feature is found [LSHD04, KLL07, EKSW10]. For example, Erdt et al. [EKSW10] use mean intensity and standard deviation of a local image patch around the landmark to decide in which direction they search for image features.

By searching along the surface normal, selection of topologically inconsistent image features occurs relatively seldom. Inconsistencies manifest themselves in surface self-intersections and flipped triangles. However, in thin regions and regions of high curvature, such triangle flips may still occur be. The biggest disadvantage of the linear search strategy is that promising features are not visible to the model. Kainmüller et al. [KLH*13] observe that this restricted visibility is especially a problem for regions with high curvature. They propose to use a ball-shaped search neighborhood for each landmark instead, which is illustrated in Figure 2.8(b). When using a ball-shaped neighborhood, the problem of surface self-intersection becomes far more critical. It is therefore crucial to use regularized image feature selection (see Section 2.4.3.3) in this case. A third search strategy has been employed by Abi-Nahed et al. [ANJY06], who determine all image features candidates in a preprocessing step. However, the robust identification of a sparse set of relevant features may be difficult for images with low contrast and greatly varying appearance.

Regardless of which local search strategy is used, the capture range of the shape model is restricted by the size of the local neighborhood. A popular way for increasing the capture range while retaining robustness is to use a hierarchical adaption scheme with a multiresolution image pyramid [CTL94]. This strategy has been widely used in many ASM implementations [HM09]. Cerrolaza et al. [CVS*12] extend this approach to the Full Multiresolution ASM, in which sparse SSMs are computed from the original fine resolution SSM via the Wavelet transform. The sparse SSMs are used for segmentation of coarse images of the image pyramid, whereas the original SSM is used for the original resolution image.

2.4.3.2. Appearance models

Arguably the simplest appearance model is to favor strongest edges in the image [CTCG95]. For example, bone structures in CT scans or radiographs have a high contrast to surrounding non-bone tissue which can be characterized well by the gradient magnitude of a voxel. The robustness can be further improved by taking the gradient direction into account and using predefined constants for normalization [WKL*01, KOE*09]. Similarly, some authors search for image intensities within a certain intensity interval [LSHD04]. Such approaches make use of the fact that CT images provide normalized intensities through the Hounsfield scale. However, even in CT images, the intensities of a certain tissue type may vary from image to image, such that the robustness of approaches based on fixed thresholds is often not guaranteed.

Most authors therefore choose a statistical approach for describing the appearance of an object at its boundary. Statistical appearance models are usually learned for each landmark separately. Training features are extracted by sampling image features from the training images at the landmark positions. If not said otherwise, it is assumed that a training example is a vector $\boldsymbol{f}_i \in \mathbb{R}^l$ and describes a one-dimensional profile as illustrated in Figure 2.8(a).

Cootes et al. [CT94] model the appearance at the boundary with a multivariate Gaussian. The negative Mahalanobis distance to the learned mean appearance $\bar{\boldsymbol{f}}$ is used to quantify the fitness of features. Note that this distribution can be learned from intensity profiles or gradient profiles. Moreover, the feature vectors $\boldsymbol{f}_i$ can be normalized, for example by dividing through $\|\boldsymbol{f}_i\|$ or $\sum_{i=1}^{l} |f_i|$. Heimann and Meinzer [HWM06] evaluate three different Gaussian appearance models for liver segmentation in CT scans. In particular, the vectors $\boldsymbol{f}_i$ contain either intensities, gradients or normalized gradients. While the best results are obtained with normalized gradient profiles in this application, Heimann and Meinzer deem the general performance of Gaussian appearance models disappointing. Although Gaussian appearance models have been outperformed by many other models for medical images, they are still quite frequently used in practice [KSG99, FRU*05].

The performance of Gaussian appearance models can be boosted by combining several Gaussians for a landmark. Toth et al. [TCR*08] learn separate Gaussians from profiles containing different features, such as gradient magnitude, Haralick inverse difference moment and Haralick entropy. During segmentation, reconstructions of the extracted features are computed by projecting them to the eigenvectors of the learned model. The quality of a set of features is then assessed by computing the combined mutual information between the features and their reconstructions, which achieves better results than using the average negative Mahalanobis distance of all Gaussians. In a latter approach [TDR*09], a weighted average Mahalanobis distance is used as fitness function. The weights are found by maximizing the correlation between the weighted Mahalanobis distance and the Euclidean distance to the landmark position. This is achieved by extracting additional training features in the vicinity of a landmark that do not lie on the organ boundary.

Instead of using a generative description of the appearance with a Gaussian or some other distribution, van Ginneken et al. [vGFS*02] follow a discriminative approach. They use a k-nearest-neighbor (kNN)-classifier to classify the sampling points on a 1D profile into the classes *inside* and *outside*. This means that in contrast to other works, the components of the vector $\boldsymbol{f}$ are not features, but classifications, such that $f_i \in [0,1]$ denotes the probability that the corresponding sampling point is inside. The fitness of $\boldsymbol{f}$ is assessed with the function

$$f(\boldsymbol{f}) = \sum_{i=1}^{\lfloor l/2 \rfloor + 1} f_i + \sum_{i=\lfloor l/2 \rfloor + 2}^{l} (1 - f_i). \tag{2.27}$$

The elements of $\boldsymbol{f}$ are ordered here from inside to outside, that is, 0 is the index of the candidate, and negative indices correspond to sampling points that lie in negative normal direction as seen from candidate coordinates. Instead of using a separate fitness function, Heimann et al. [HMMW07] propose to classify feature vectors $\boldsymbol{f}$ directly with a moderated kNN-classifier by assigning the classes *boundary* and *non-boundary* instead of *inside* and *outside*. As fitness function, they use the probability that a feature is on the boundary. Whenever a discriminative approach is chosen, both positive and negative training examples must be available. This is usually done by sampling several features in each training image in a landmark's vicinity, and classifying them based on the label in the corresponding expert segmentation.

Some authors also combine statistical information with explicit rules. For example, Kainmueller et al. [KLL07] estimate the image intensity range of liver and liver tumor tissue in a contrast enhanced CT scan by fitting a Gaussian Mixture Model to the intensity histogram of the scan. A rule-based scheme is used to identify the mixture components corresponding to the liver and tumor tissue, respectively.

In general, which appearance model works best is highly dependent on the image modality and the organ. New appearance models can be easily constructed by combining some of the building blocks presented above: One can choose different image features, different normalization methods, and finally select a fitness function to assess the quality of features. Statistical appearance models have the fundamental advantage that they can be easily adapted to new applications because they learn relevant parameters directly from the training data. However, the good performance of rule-based models such as the liver tissue model of Kainmueller et al. [KLL07] shows that 'handcrafting' models to a specific application can be beneficial. The risk of these methods is that one might design rules that work well on the available test set, but do not generalize when applied to new data.

2.4.3.3. Image feature selection

The local image feature search computes a set of candidate positions around each landmark. The appearance model is used to assign a score to each candidate such that they can be ranked. Finally, one of the candidates is chosen as new position for the landmark. The obvious strategy which is employed in most ASM implementations is to select the candidate with the highest score [CTCG95]. However, the appearance model does not necessarily assign the best score to the image feature that corresponds to the boundary. Such outliers of the appearance model may degrade the robustness of the ASM, especially if they have a large distance to the true boundary. This is because most approaches fit the shape model to the image features using least squares when imposing shape constraints (see Section 2.4.4). The smoothing effect of least squares draws also landmarks that are adjacent to an outlier landmark towards the wrong image features. A simple remedy is to add a distance penalty to the score of the appearance model in order to penalize the distance of a feature to the current landmark position [WKL*01]. This has the effect that the selection of candidates that cause large displacements becomes unlikely. Beyond that, there exist three more complex strategies that tackle the problem:

1. Identify and correct outliers of the appearance model.
2. Reduce the influence of outliers.
3. Select a consistent set of image features.

An example for the first strategy is the approach of Lekadir et al. [LMY07]. They propose a pose invariant shape descriptor that describes spatial relations between triplets of landmarks with a single number. For each triplet, confidence intervals are learned. During segmentation, outlying landmarks are iteratively determined by identifying those landmarks whose corresponding triplets are outside the learned confidence intervals. The position of these landmarks are then corrected without taking further image information into account. The outlier information is also integrated into the next image feature search, such that an outlier is not chosen twice.

Instead of correcting the position of outliers, one can try to remove their influence. For example, weighted least squares can be used instead of least squares backprojection when enforcing shape constraints (see Section 2.4.4). Graham and Rogers [RG02] propose two alternative methods for determining the weights: The first method is to assign for each landmark a weight, which either depends on the landmark's residual $\boldsymbol{r}^i$, or on the score of the appearance model. The second method, which showed superior performance in their evaluation, is to use only a subset of the landmarks during backprojection, which can be achieved with binary weights. In particular, weighted backprojection with randomly sampled landmarks is repeated several times and the most consistent parameters are chosen, for example those parameters that achieve the median residual magnitude $\|\boldsymbol{r}\|$.

The aforementioned strategies select the image feature with the highest score, but try either to suppress outliers or to reduce their influence. An elegant alternative is to select a set of features that is consistent over the complete boundary in the first place. Consistency constraints are achieved by adding penalty terms that depend on two adjacent landmarks. The penalty terms of the works discussed here aim at enforcing smoothness of the detected boundary by penalizing the difference between displacement vectors of adjacent landmarks. To the best of my knowledge, the first approach along that line has been proposed by Behiels et al. [BMVS02], who use a minimum cost path algorithm to determine a smooth boundary in a 2D ASM for radiograph segmentation. Unfortunately, the method does not extend to 3D shapes. The first 3D approach has been proposed by Heimann et al. [HMMW07], who adopt a more general segmentation algorithm called *optimal surface detection* to ASMs. They construct a directed graph, where each candidate position corresponds to a node. Candidates belonging to the same landmark are connected by edges which point from the outside to the inside of the boundary. Additionally, inter-landmark edges are added that connect candidates of adjacent nodes. Smoothness constraints are integrated by creating only inter-landmark edges between candidates that move landmarks in similar directions. An optimal solution is determined by solving a max-flow problem with the algorithm of Boykov et al. [BK04]. A drawback of Heimann's approach is that the smoothness constraints do not use physical coordinates, but are based on the positions of the candidates on the linear search profile. Thus, a boundary that is deemed smooth by Heimann's approach may not be smooth in a physical sense. Kainmüller et al. [KLH*13] construct a Markov Random Field for selecting consistent image features. In their approach, physical smoothness constraints can be readily incorporated. A second advantage is that the approach is not limited to the linear feature search strategy. On the other hand, they report rather long computation times for calculating an optimal state of the Markov Random Field for large, ball-shaped search neighbourhoods.

2.4.4. Imposing shape constraints

While outlier detection methods avoid that neighboring landmarks move inconsistently, they do not take the global object shape into account. By imposing shape constraints with an SSM, one can ensure that the shape of the segmentation is similar to the training shapes. Thus, leaking of the object into neighboring structures can—at least partially—be avoided. In the following, different techniques for imposing shape constraints are presented. If not said otherwise, it is assumed that the shape model is a linear model, because linear models are used by most ASMs. Techniques for imposing shape constraints using nonlinear shape models are discussed briefly in Section 2.4.4.4.

It is important to remember that the SSM has been learned in Procrustes space (see Section 2.2.1), while the deformed shape is in the coordinate system of the image. In order to impose shape constraints, *pose parameters* must be computed which define an affine transformation between these two coordinate systems. The pose parameters consists of a scale factor, one or more rotation angles[5] and a translation vector. Usually, pose parameters are handled as external parameters. There exist closed form solutions both in 2D [CTCG95] and in 3D [Hor87] for computing pose parameters that superimpose the deformed shape $\hat{\boldsymbol{x}}$ on the mean shape $\bar{\boldsymbol{x}}$ optimally in a least squares sense.

2.4.4.1. Standard ASM

The most frequently used approach for constraining shapes has been proposed by Cootes et al. [CTCG95] in their seminal papers on SSMs. It works as follows:

1. Pose parameters are estimated, and the deformed shape $\hat{\boldsymbol{x}}$ is mapped from the image to the model coordinate system.
2. The deformed shape is projected into the principal subspace using the equation

$$\boldsymbol{b} = \boldsymbol{P}^T(\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}). \tag{2.28}$$

   If weights are available that assess the quality of the image features , as it is done for example in the Robust ASM search of Rogers and Graham [RG02], the equation is

$$\boldsymbol{b} = (\boldsymbol{P}^T\boldsymbol{W}\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{W}(\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}), \tag{2.29}$$

   where $\boldsymbol{W} \in \mathbb{R}^3$ is a diagonal matrix containing the weights.
3. Constraints are imposed bounds on the principal components b_i, usually by enforcing that

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \text{ for all } i = 1, \ldots, t. \tag{2.30}$$

   Note that under the assumption of a Gaussian shape distribution, the b_i are independently distributed, and each b_i is Gaussian distributed with zero mean and standard deviation $\sqrt{\lambda_i}$. Thus, the constraints enforce that b_i can move at most three standard deviations away from its expected

---

[5] One rotation angle suffices for 2D shapes. For 3D shapes, three rotatation angles are required.

value. An alternative strategy to constrain $\boldsymbol{b}$ is to rescale the vector such that that $M(\boldsymbol{b}) \leq D_{\max}$, where

$$\mathrm{M}(\boldsymbol{b}) = \sum_{i=1}^{t} \frac{b_i^2}{\lambda_i} \tag{2.31}$$

is the Mahalanobis distance in the principal subspace, and $D_{\max}$ is a predefined threshold.

4. From the constrained shape parameters, a plausible shape is generated by computing

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b}. \tag{2.32}$$

5. The constrained shape $\boldsymbol{x}$ is mapped back to the image coordinate system using the estimated pose parameters.

It is also possible to iterate several times between pose and shape parameter estimation in order to improve the results. In that case, pose parameters are initially estimated using the mean shape, and later using the SSM instance computed in the previous iteration.

The vast majority of ASM implementations constrain shapes as outlined above. While this technique is very fast and easy to implement, it is also one reason for the limited flexibility and delineation accuracy of ASMs. This is because it partitions the shape space into two classes: Allowed shapes and disallowed shapes. Allowed shapes are a subset of the set of all linear combinations of the training shapes. They are necessarily elements of the affine PCA subspace, and must additionally be close to the mean. Conversely, any shape that can not be expressed as a linear combination of the training shapes is disallowed. However, one can not expect that all possible shape variations are present in the training shapes, especially in medical imaging, where often few examples are available. Even if a large amount of training data was available, one typically wants to reduce the dimensionality in order to save memory and retain efficiency. The covariance matrix of 3D landmark vectors with a few thousands landmarks has several million entries. But dimensionality reduction, if it is implemented with Equations 2.28 and 2.32, leads again to a restriction of shapes to a small linear subspace.

2.4.4.2. Shape Constrained Deformable Models

Shape Constrained Deformable Models (SCDMs) are a variant of the standard ASM which has been proposed by Weese et al. [WKL*01]. The main difference is that SCDMs constrain shapes by minimizing the energy

$$E(\boldsymbol{x}, \boldsymbol{b}) = \alpha E_{\text{ext}}(\boldsymbol{x}) + E_{\text{int}}(\boldsymbol{x}, \boldsymbol{b}). \tag{2.33}$$

The external energy $E_{\text{ext}}(\boldsymbol{x})$ draws the model mesh towards detected image feature, while the internal energy $E_{\text{int}}(\boldsymbol{x}, \boldsymbol{b})$ enforce similarity to the training shapes. The free parameter α is used to balance both energies. This parameter must be determined for each application separately.

The external energy[6] is defined by

$$E_{\text{ext}}(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \left(\left(\frac{\boldsymbol{G}(\hat{\boldsymbol{x}}^i)}{\|\boldsymbol{G}(\hat{\boldsymbol{x}}^i)\|} \right)^T (\hat{\boldsymbol{x}}^i - \boldsymbol{x}^i) \right)^2 . \tag{2.34}$$

The w_i are weights assigned by the appearance model, and $\boldsymbol{G}(\hat{\boldsymbol{x}}^i)$ denotes the image gradient at the feature position $\hat{\boldsymbol{x}}^i$. By projecting the displacement $\hat{\boldsymbol{x}}^i - \boldsymbol{x}^i$ on the normalized image gradient, image features that are perpendicular to the image gradient have lower energy.

The internal energy takes the form

$$\begin{aligned} E_{\text{int}}(\boldsymbol{x}, \boldsymbol{b}) &= \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} \|(\boldsymbol{x}^i - \boldsymbol{y}^i(\boldsymbol{b})) - (\boldsymbol{x}^j - \boldsymbol{y}^j(\boldsymbol{b}))\|^2 & (2.35) \\ &= \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} \|\boldsymbol{r}^i - \boldsymbol{r}^j\|^2 & (2.36) \end{aligned}$$

with $\boldsymbol{y}(\boldsymbol{b}) = \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b}$ and $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}(\boldsymbol{b})$, and $\mathcal{N}(i)$ is the set of indices of the neighbors of landmark i. The energy considers all shapes within the principal subspace as plausible and assigns zero energy to them. Hard bounds depending on the eigenvalue as in Equation 2.30 are not used. Moreover, shapes may leave the principal subspace, but thereby generate a nonzero energy. This energy is smaller if neighbouring landmarks move in similar direction. Thus, the internal energy can be seen as a regularization that maintains a smooth model mesh whenever it leaves the principal subspace.

For minimizing the energy, Weese et al. [WKL*01] use the conjugate gradient method. Although the problem is defined in the high-dimensional space $\mathbb{R}^{dn}$, the optimization is fast because the minimization amounts to solving a sparse linear system.

SCDMs and its variations have been applied to various segmentation task [WKL*01, HMMW07, EPS*08, KOE*09]. Its internal energy can also be simplified such that only a fixed template mesh is required which replaces $\boldsymbol{y}(\boldsymbol{b})$ in Equation 2.35.

2.4.4.3. Other energy minimization based shape constraints

Several other works also use energy minimization for enforcing shape constraints in ASM segmentation. Gollmer and Buzug [GB12] use the DFFS (see Section 2.2.2.3), $\|\boldsymbol{r}\|^2$, as shape energy. This approach is actually very similar to the internal energy of SCDMs: It also assigns zero energy to shapes within the principal subspace. However, as only the magnitude of the residual $\boldsymbol{r}$ is penalized, the resulting segmentation contour may not to be smooth. Abi-Nahed et al. [ANJY06] align the shape model to a set of features that have been extracted from the image in a preprocessing step. Their approach

---

[6] The external energy presented in the work of Weese et al. [WKL*01] is slightly different to that in Equation 2.34, because they conduct image feature search from triangle centers, not from mesh points. This difference is ignored here to simplify the presentation.

is a combination of ASMs and Robust Point Matching, a point set registration algorithm which estimates outliers during registration. Additional forces are added to the optimization to regularize the transformation.

2.4.4.4. Shape constraints in nonlinear models

Energy minimization has also been used to constrain shapes in Active Shape Models with nonlinears SSMs (see Section 2.2.4). Cootes et al. [CT97] use the log-likelihood of a GMM density as shape energy. Twining and Taylor [TT01] use a ‘proximity to data’ measure derived from Kernel PCA for the same purpose.

In both cases, a deformed shape $\hat{x}$ is constrained by minimizing the shape energy until a predetermined plausibility threshold ξ is reached. In contrast to the energy minimization approaches for linear models presented above, no data term is used, such that no force ensures that the optimized shape remains close to the detected image features $\hat{x}$. A second disadvantage is that the threshold ξ must be chosen. There is no obvious choice for this parameter. Furthermore, it leads to a hard separation between plausible and implausible shapes.

2.4.5. Post-processing

Instead of relaxing the shape constraints, some authors prefer to postprocess the ASM segmentation with a second method. For example, Kainmüller et al. [KLL07] use a *constrained free form deformation* to improve the segmentation. Their methods works on a mesh that has been computed from the ASM’s segmentation. For each landmark, three forces are balanced: One force draws a landmark towards a detected image feature, one enforces smoothness of the boundary, and the third force ensures that the landmark remains close to the final ASM segmentation. Erdt et al. [EKSW10] use a regularization similar to that of SCMDs (Equation 2.35) with a fixed template mesh, but additionally use local curvature features that control the local elasticity of the free form deformation. Makni et al. [MPL*09] use a Markov Random Field to model the image intensities. They use an SSM-based segmentation as an initialization for the Iterated Conditional Modes algorithm to compute a labeling for which the energy of the Markov Random Field takes a local minimum.

A general drawback of such hybrid post-processing methods is that they make the model used for segmentation more complex. Additional model assumptions are introduced that may be in conflict to those assumptions made in previous stages of the segmentation. Often, several additional parameters are introduced that must be tuned, which makes the adaption to new applications harder.

2.4.6. Discussion of Active Shape Model segmentation

The ASM is the most widely used algorithm for medical image segmentation with landmark-based shape models. The fact that the ASM is a local search algorithm and therefore requires a good initialization is nowadays no longer a big disadvantage, because machine learning based object detection can provide robust initializations within seconds. Beside the initialization, the performance of the ASM depends on the employed search strategy, the local appearance model that describes the texture at the object's boundary, as well as the way shape constraints are enforced.

Which appearance model performs well depends strongly on the underlying application. Although statistically based appearance models are more generic than application-specific models, good 'handcrafted' models show often comparable or even better results. A reason for this might be that if few training instances are available, statistical models can not capture the statistics of the boundary's appearance sufficiently well.

The current linear search strategy of the ASM has the problem that it has a limited visibility field, that is, it may overlook important image features. More complex strategies, such as the ball-shaped search strategy of Kainmüller et al. [KLH*13], have a larger visibility field, but come at the cost of considerably higher computation times. A possible solution to this visibility problem is outlined in the outlook of this thesis in Section 7.2.3, where the concept of a symmetric multi-shape model adaption is presented.

Arguably the biggest disadvantage of the standard ASM is how shape constraints are enforced: Shapes are constrained to be certain linear combinations of the training shapes. This approach is too restrictive to achieve a high segmentation accuracy on previously unseen images. Less restrictive variants of the ASM, such as SCDMs [WKL*01], achieve a higher flexibility at the cost of limited specificity. As discussed in detail in Section 5.6.2, SCDMs deem shapes as plausible that are arbitrarily far away from the mean. Postprocessing ASM-based segmentations with different algorithms comes at the cost of increasing the model complexity and introducing additional parameters. Finally, current ASM variants do not provide a unified treatment of linear and nonlinear shape priors. This point, as well as the limited flexibility of the standard ASM, is addressed by the Probabilistic ASM presented in Chapter 5.

2.5. Alternatives to Active Shape Models

In this section, segmentation algorithms are reviewed that—like the ASM— also enforce global shape constraints with SSMs. In medical imaging, the Active Appearance Model (AAM), which is discussed in Section 2.5.1, is probably the most popular alternative to the ASM. The *Diffusion Snakes* of Cremers et al. [CTWS02, CKS03] discussed in Section 2.5.2 segment images by contour evolution. The shape energies of Diffusion Snakes are also used in the Probabilistic ASM devised in this work.

In addition to these approaches, Wang and Staib [WS00] use gradient descent to fit the shape model to an edge image. They use the Mahalanobis Distance in PCA subspace (Eq. 2.31) as shape energy. They also include priors on the pose parameters in their framework.

2.5.1. Active Appearance Models

Active Appearance Models [CET01] describe both global shape and texture variation as well as correlations between them. From a set of training images, annotated by landmark vectors, *shape free image* *patches* are computed by warping the image to the mean shape. The shape free image patches are sampled with a fixed sampling pattern in order to obtain a texture vector $\boldsymbol{g}_i$ for each training example. By applying PCA separately to shape and texture vectors, one obtains two linear models

$$\begin{aligned} \boldsymbol{x} &= \bar{\boldsymbol{x}} + \boldsymbol{P}_{\text{sh}}\boldsymbol{b}_{\text{sh}} \quad \text{and} && (2.37) \\ \boldsymbol{g} &= \bar{\boldsymbol{g}} + \boldsymbol{P}_{\text{tx}}\boldsymbol{b}_{\text{tx}}. && (2.38) \end{aligned}$$

Shape and appearance parameters are usually not directly comparable because they are measured in different units. A weight matrix $\boldsymbol{W}_{\text{sh}}$, which can be estimated from the training data, accounts for the difference. A third PCA on the combined shape and texture parameters $\boldsymbol{b}^T = ((\boldsymbol{W}_{\text{sh}}\boldsymbol{b}_{\text{sh}})^T \,|\, \boldsymbol{b}_{\text{tx}}^T)$ results in the final linear model

$$\boldsymbol{b} = \boldsymbol{P}\boldsymbol{c}. \tag{2.39}$$

The vector $\boldsymbol{c}$ contains the *appearance parameters*. Additional parameters required for segmentation are the pose parameter $\boldsymbol{t}$ and lightning parameters $\boldsymbol{u}$ that account for differences in global illumination.

AAMs segment an image by changing the model parameters $\boldsymbol{h}^T = (\boldsymbol{c}^T \,|\, \boldsymbol{t}^T \,|\, \boldsymbol{u}^T)$ so as to minimize the difference

$$\delta I(\boldsymbol{h}) = I_{\text{im}}(\boldsymbol{h}) - I_{\text{md}}(\boldsymbol{h}) \tag{2.40}$$

between image intensities $I_{\text{im}}(\boldsymbol{h})$ and the intensities $I_{\text{md}}(\boldsymbol{h})$ generated by the current model parameters. This is done with gradient descent using the first order Taylor expansion

$$\delta I(\boldsymbol{h} + \delta\boldsymbol{h}) = I(\boldsymbol{h}) + \frac{\partial I(\boldsymbol{h})}{\partial \boldsymbol{h}}\delta\boldsymbol{h}. \tag{2.41}$$

Because calculating the derivative matrix $\frac{\partial I(\boldsymbol{h})}{\partial \boldsymbol{h}}$ is time consuming, it is replaced in AAMs by a fixed, precomputed matrix $\boldsymbol{R}$ which is estimated by numeric differentiation on the training examples.

Several improvements of the basic AAM have been proposed: For example, the constrained AAM by Cootes et al. [CT01] embeds AAMs into a probabilistic framework, which allows to specify priors on the model parameters $\boldsymbol{p}$. Matthews and Baker [MB04] discuss several variations of the AAM and propose the *inverse compositional* fitting algorithm. AAMs have also been applied to various 3D medical image segmentation tasks, for example heart segmentation in MRI [SP05] or triplane echocardiograms [HULR07], brain segmentation in MRI [BCT*08] and prostate segmentation in transrectal ultrasound and MRI scans [GOM*10], among others.

Studies that directly compare AAMs and ASMs are rare. Cootes et al. [CT04] evaluate both AAMs and ASMs on a 2D face and a 2D brain data set. The ASM has a slightly worse capture range than the AAM on the face data, but a considerably better capture range on the brain data. The point location accuracy of the ASM is better in both applications. Finally, the ASM converges considerably faster. It is hard to say whether these results generalize to other applications. At least one can say that ASM is the faster algorithm.

There are several other arguments in favor of the ASM:

1. The gap between the number of training examples and their dimensionality becomes even larger for the AAM, because both shape and texture variation are modeled.
2. One of the greatest drawbacks of the ASM, its lack of flexibility, is also a weakness of the AAM. It is even likely that the problem is more severe for the AAM, because it does not only restrict shape parameters, but also texture parameters to a linear subspace.
3. The AAM search can be impeded by the assumed linear relation between shape and texture used during AAM segmentation (see Equation 2.39).
4. The AAM search can be impeded by pathologies in medical images. Pathologies such as large tumors usually have an even greater impact on the texture than on the shape.
5. AAMs have a huge memory consumption, in particular when applied to 3D images [HM09].

2.5.2. Diffusion Snakes

Cremers et al. [CTWS02] propose *Diffusion Snakes*, a framework which combines variational Mumford-Shah segmentation with shape priors. In variational Mumford-Shah segmentation, one seeks a function U that approximates the image I and is piecewise smooth, but may have discontinuities at some contour C. Cremers et al. model the contour by a spline with a fixed number of control points which correspond to the landmarks of the shape. Statistics over the set of shapes is incorporated into the framework by defining the combined functional

$$E_{\text{DS}}(U,C) = E_{\text{MS}}(U,C) + \alpha E_{\text{shape}}(C), \tag{2.42}$$

where $E_{\text{MS}}(U,C)$ is an image force that corresponds to the original Mumford-Shah functional, $E_{\text{shape}}(C)$ is the shape energy and α a balancing factor. Cremers et al. [CTWS02] use the Mahalanobis distance as a shape energy (Equation 2.10). In a later publication, KPCA is used to define nonlinear shape priors [CKS03]. The functional is minimized by evolving the contour along the normals of the control points using gradient descent.

In the Diffusion Snakes framework, pose parameters are integrated into the minimization. This is in contrast to ASMs, which treat pose parameters as external parameters. The idea is to define a pose invariant shape energy $\tilde{E}_{\text{shape}}(\boldsymbol{x}) = E_{\text{shape}}(\tilde{\boldsymbol{x}})$, where $\tilde{\boldsymbol{x}}$ is a shape that is aligned with the mean shape. It is assumed that the mean shape is centered, which means that its center of gravity is the origin, and is scaled to unit length. A shape $\boldsymbol{x}$ is aligned with the mean by computing

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{R}\boldsymbol{x}_c}{||\boldsymbol{R}\boldsymbol{x}_c||}, \tag{2.43}$$

where $\boldsymbol{R}$ is a rotation matrix and $\boldsymbol{x}_c$ is the centered version of $\boldsymbol{x}$. During optimization, the derivative of the pose invariant shape energy $\tilde{E}_{\text{shape}}(\boldsymbol{x})$ is computed with the chain rule. The disadvantage of integrating pose parameters into the optimization is that the Jacobian of Eq. 2.43 is not sparse. Handling large, dense matrices slows done the computation time considerably and requires much memory. As an example, the dimension of the optimization problem for a 3D shape model with 2000 landmarks is 6000, which means that the Jacobian would have 36 million entries.

The original publications on Diffusion Snakes [CTWS02, CKS03] give exemplary segmentation results on 2D images, but do not provide a quantitative evaluation of the segmentation accuracy, nor do they specify run times. Recently, Tejos et al. [TICB09] presented an extension of Diffusion Snakes to 3D images which uses a triangulated mesh instead of splines to represent the segmentation boundary. Also in this case, a quantitative evaluation and information about the running time of the method is missing. In general, one can expect that Diffusion Snakes are considerably slower than ASMs in 3D, because the latter algorithm does not need to maintain huge matrices in order to achieve pose invariance.

2.6. Quantification of segmentation accuracy

Quantitative evaluation of segmentation algorithms is of major importance in medical imaging. It allows for an objective comparison of different approaches and their performance on various organs and image modalities. The evaluation is usually done by comparing the segmentation of an algorithm with a reference segmentation. As it is in most cases impossible to obtain a ground truth segmentation for an image, reference segmentations are created manually by human observers with sufficient anatomical knowledge, for example by radiologists or their assistants.

Due to noise, partial volume effects and low contrast, manual segmentation of organs is subject to inter- and intra-observer variability. These issues can be addressed by combining several manual segmentations of the same organ. To this end, the STAPLE algorithm [WZW04] has been proposed which estimates the performance of different raters and weighs their segmentations accordingly when combining them to a single reference segmentation.

In the following, several measures are formally introduced that have been used within this work to compare an algorithmic segmentation with a reference segmentation. Most definitions presented here are based on those given in the evaluation study of Heimann et al. [HvGS*09] on liver segmentation algorithms. Additionally, Dice coefficient [Dic45] and Jaccard index [Jac01] are defined, as they are used in many studies instead of the equivalent volumetric overlap error. In the following, the sets of voxels of two segmentations are denoted by $\mathcal{A}$ and $\mathcal{B}$. Moreover, the sets $\mathcal{A}_{\text{surf}} \subseteq \mathcal{A}$ and $\mathcal{B}_{\text{surf}} \subseteq B$ contain the surface voxels of $\mathcal{A}$ and $\mathcal{B}$, respectively. Surface voxels have at least one non-object voxel in their 18-neighborhood. One can differentiate between two kinds of measures: Volumetric measures and surface-based measures. The latter is computed from the set of surface voxels only.

Volumetric measures. Typical volumetric measures are *Dice coefficient*, *Jaccard index* and *volumetric overlap error*. The Dice coefficient is defined by

$$\mathrm{Dice}(\mathcal{A},\mathcal{B}) = \frac{2|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}|+|\mathcal{B}|}. \tag{2.44}$$

The Jaccard index is the ratio between the size of the intersection and the union of the sets of voxels, that is

$$\mathrm{Jaccard}(\mathcal{A},\mathcal{B}) = \frac{|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}\cup\mathcal{B}|}. \tag{2.45}$$

Both Dice coefficient and Jaccard index take values between 0 and 1, where 0 means that $\mathcal{A}$ and $\mathcal{B}$ are disjoint, and 1 implies perfect overlap, that is $A = B$. Conversely, the volumetric overlap error

$$\mathrm{VOE}(\mathcal{A},\mathcal{B}) = 100\cdot(1-\frac{|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}\cup\mathcal{B}|}) \tag{2.46}$$

is 0% if and only if $\mathcal{A} = \mathcal{B}$, and 100% if and only if $\mathcal{A}$ and $\mathcal{B}$ are disjoint.

Dice coefficient, Jaccard index and VOE are equivalent, that is, one measure can be directly computed from the other. For that reason, it suffices to specify only one of the three measures. In the evaluation and application sections, the Dice coefficient is used because it is arguably the most popular volumetric measure in the medical imaging community. The equivalence of Jaccard index and VOE follows directly from their definitions. Moreover, it holds that

$$\mathrm{Dice}(\mathcal{A},\mathcal{B}) = \frac{2\cdot\mathrm{Jaccard}(\mathcal{A},\mathcal{B})}{1+\mathrm{Jaccard}(\mathcal{A},\mathcal{B})}. \tag{2.47}$$

The *relative volume difference* is an asymmetric measure that indicates whether a segmentation over- or undersegments a structure. It is defined by

$$\mathrm{RVD}(\mathcal{A},\mathcal{B}) = 100\cdot\frac{|\mathcal{A}|-|\mathcal{B}|}{|\mathcal{B}|}. \tag{2.48}$$

Surface based measures. Surface based measures compare two segmentation boundaries. They are all based on the minimal distance

$$\mathrm{dist}(b,\mathcal{A}) = \min_{a\in\mathcal{A}_{\mathrm{surf}}} \|a-b\| \tag{2.49}$$

of a voxel b to the set of surface voxels of $\mathcal{A}$.

Both *average symmetric surface distance*

$$\mathrm{ASD}(\mathcal{A},\mathcal{B}) = \frac{1}{|\mathcal{A}_{\mathrm{surf}}|+|\mathcal{B}_{\mathrm{surf}}|}\left(\sum_{a\in\mathcal{A}_{\mathrm{surf}}}\mathrm{dist}(a,\mathcal{B}) + \sum_{b\in\mathcal{B}_{\mathrm{surf}}}\mathrm{dist}(b,\mathcal{A})\right) \tag{2.50}$$

and *root mean square symmetric surface distance*

$$\mathrm{RMS}(\mathcal{A},\mathcal{B}) = \sqrt{\frac{1}{|\mathcal{A}_{\mathrm{surf}}|+|\mathcal{B}_{\mathrm{surf}}|}\sum_{a\in\mathcal{A}_{\mathrm{surf}}}\mathrm{dist}^2(a,\mathcal{B}) + \sum_{b\in\mathcal{B}_{\mathrm{surf}}}\mathrm{dist}^2(b,\mathcal{A})} \tag{2.51}$$

average the minimal distance of all surface voxels of one segmentation to the other, and vice versa. The main difference is that by squaring, the RMS weighs larger errors more strongly. Finally, the *Hausdorff distance* quantifies the maximal error of the segmentation. It is defined by

$$\mathrm{HD}(\mathcal{A},\mathcal{B}) = \max\left(\max_{a\in\mathcal{A}_{\mathrm{surf}}} \mathrm{dist}(a,\mathcal{B}), \max_{b\in\mathcal{B}_{\mathrm{surf}}} \mathrm{dist}(b,\mathcal{A})\right). \tag{2.52}$$

In this work, all surface based error measures are specified in millimeter.

In context of shape modeling, in which the segmentation is available as a 3D mesh, it is also popular to restrict the computation of surface error measures to the points of the model mesh that corresponds to the final segmentation [WKL*01, KOE*09].

2.7. Further applications of shape models

Beside image segmentation, SSMs have several other applications in medical imaging. In *surface extrapolation*, SSMs are used to extrapolate the complete shape of a structure from a sparse set of points that are acquired during surgery using a 3D localizer system [FLJ99,RST*07]. Related to surface extrapolation are 2D-3D registration approaches, where the 3D shape of an object is estimated from several X-ray images [FL99, DLvB*10]. In *shape analysis*, the effect of diseases on the shape of an organ is studied. These studies investigate the differences in shape of certain organs in different groups of patients; for example, the shapes of brain structures of patients suffering from Alzheimer [FOP*07] or shizophrenia [SLPG04] are compared to those of control groups. SSMs have also been used to model patient specific hearing aids [UNSF08].

Outside the medical imaging community, SSMs are popular for face modeling: Blanz and Vetter [BV99] propose *morphable face models*, which model both shape and texture of 3D faces. These models can be adapted to 2D images and used to synthesize facial expressions of persons in both 2D and 3D. Morphable face models have also been applied to face recognition [BV03]. *Morphable body models* have been used for editing the shape of people in video sequences [JTST10].

Some of the techniques presented in this thesis, in particular the methods proposed for establishing correspondence, can be used in these applications as well.

2.8. Summary and discussion

This chapter introduced the current state-of-the-art of model-based image segmentation using the landmark representation. Based on this review, this section identifies some major challenges in the research field and explains how the new methods devised for this thesis address these challenges.

Although various methods have been proposed for establishing correspondence, there is still no general solution to the problem available. The theoretically most well founded approach to the correspondence problem so far is groupwise optimization. In practice, groupwise optimization is computationally very expensive, especially in 3D. The methods devised in this work therefore focus on providing an initialization close to a ‘good’ local optimum in order to decrease the computational burden. Because, almost all 3D groupwise optimization approaches are restricted to shapes of spherical topology, a new algorithm based on planar consistent parameterization is presented in this thesis, which optimizes a groupwise objective function for shapes of genus 1.

An alternative approach to groupwise optimization is to integrate group information into pairwise registration-based model building using shape similarity trees. To this end, this thesis provides a detailed comparison of several approaches for computing the tree, using different data sets and different pairwise correspondence algorithms. The latter include a new algorithm based on nonrigid mesh registration, which is considerably faster than groupwise optimization, and also much more versatile, because it is not restricted to shapes of a particular topology.

The second part of this thesis addresses the challenge of accurate shape model-based medical image segmentation. The arguably most fundamental disadvantage of the standard ASM for segmentation is its limited flexibility. Different approaches have been taken in order to address this problem: Some researchers increase the flexibility during the training phase, for example by adding artificial training examples [KTL*08], or by modifying the covariance matrix in order to add additional, smooth deformations [CT95]. Such approaches require additional model assumptions and thus increase the model complexity. Moreover, they are governed by free parameters whose optimal choice is application-dependent, which makes the adaptation to other applications harder. An alternative approach is to increase the flexibility during segmentation: Post-processing of segmentations (e.g. [KLL07]) or hybrid segmentation approaches (e.g. [EKSW10]) again increase the model complexity and introduce additional parameters. SCDMs [WKL*01] use only smoothness as additional model assumption, but the increased flexibility comes at the cost of a lower specificity. Moreover, none of the ASM variants proposed is able to seamlessly exchange linear shape priors with nonlinear ones.

The Probabilistic ASM proposed in this thesis addresses the problems of limited flexibility and specificity. Although it has only one additional model assumption, namely smoothness of organ boundaries, it constrains shapes in a more flexible way than the standard ASM. Because it explicitly models shape variation with probability distributions, it is more specific than SCDMs. Finally, it also allows to integrate nonlinear shape priors, which facilitates a direct comparison of linear and nonlinear shape models for medical image segmentation.

3. Establishing surface correspondence

Establishing point correspondence is a necessary prerequisite for modeling shape variation with landmark-based shapes. This chapter presents several new methods for establishing correspondence for groups of training shapes. The methods shown here allow the construction of shape models for a large variety of different organs.

Most methods discussed in this chapter are parameterization methods which map a set of training shapes consistently to a common parameter domain. Once the parameterizations have been constructed, landmark vectors can be computed by sampling on each parameterization with same sampling pattern. An advantage of using parameterization-based methods is that they inherently guarantee that the topology of the input shapes is preserved, and no triangle flips or foldings occur in the generated landmark vectors. This topology preservation can not be trivially achieved by using mesh registration alone. Furthermore, consistent parameterization provides a starting point for groupwise optimization (see Section 2.3.3), which is arguably the mathematically most rigorous solution to the correspondence problem proposed so far.

After some necessary terminology has been introduced in Section 3.1, Section 3.2 presents consistent parameterization methods for shapes with spherical topology, and Section 3.3 describes a parameterization method for shapes of genus 1. Spherical parameterization is especially important in context of medical imaging, because many organ shapes have spherical topology, such as the liver, the kidneys, the lungs, the ventricles of the heart, the spleen, and the prostate. Genus 1-parameterization is applicable to shapes that have a single hole, such as vertebrae. Both Section 3.2 and Section 3.3 include the description of an algorithm for groupwise optimization on the respective parameter domain.

Usually, landmark vectors are reconstructed from the parameter domain using uniform sampling, but in some cases, non-uniform sampling is desired. Adaptive sampling allows for example to compensate for distortions in the parameterizations, or allows to use more sampling points in high curvature regions of the shapes. A generic approach to create adaptive sampling patterns for different parameter domains is presented in Section 3.4. It is applicable to both the spherical and the planar parameter domain.

Section 3.5 presents an algorithm for constructing shape models based on pairwise nonrigid mesh registration. In contrast to the parameterization-based approaches, it has the advantage that it is independent of the topology of the shape, and is therefore applicable to a much broader set of shape classes. The nonrigid mesh registration algorithm works also well for shapes with large intra-class variability, because it is more flexible than the rigid or affine registration algorithms that are used by many authors for establishing correspondence. Moreover, it handles thin, elongated structures well, such as the pancreas. The pancreas is—despite its spherical topology—not well suited for spherical parameterization because its thin shape becomes strongly distorted when it is mapped to the sphere.

The chapter closes in Section 3.6 with a discussion of several methods for constructing shape similarity trees, which are used to improve the performance of pairwise correspondence methods. They can be combined with the nonrigid mesh registration algorithm and with all consistent parameterization methods presented in this work, as all of these methods are inherently pairwise.

3.1. Preliminaries

The following terminology and notation is used within this chapter: A shape $\mathcal{S} \subset \mathbb{R}^3$ is a continuous two-dimensional surface embedded in 3D space. A shape $\mathcal{S}$ can be approximated by a triangle mesh M. A mesh $M = (\mathcal{V}, \mathcal{F})$ is defined by a set of m points $\mathcal{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_m\}$, $\boldsymbol{v}_i \in \mathbb{R}^3$ for $i = 1, \dots, m$, and a set of triangles $\mathcal{F} = \{t_1, \dots, t_{|\mathcal{F}|}\}$, $t_j \subset \{\{a, b, c\} \mid a, b, c \in \{1, \dots, m\}\}$ for $j = 1, \dots, |\mathcal{F}|$. A mesh can also be considered as a graph. The set of nodes of the undirected graph G_M induced by the mesh $M = (\mathcal{V}, \mathcal{F})$ corresponds to the mesh's set of points $\mathcal{V}$. The edge $\{i, j\}$ is an edge of G_M if and only if there is a triangle $t \in \mathcal{F}$ such that $i, j \in t$. Furthermore, a triangle mesh M also induces a neighborhood relation. The points $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}$ are called *neighbors* if there is a triangle $t \in \mathcal{F}$ such that $i, j \in t$. The set

$$\mathcal{N}(i) = \{j \mid \exists t \in \mathcal{F} \text{ such that } i, j \in t\} \tag{3.1}$$

for $i \in \{1, \dots, m\}$ contains exactly the indices of all neighbors of point $\boldsymbol{v}_i$.

A *parameterization* of a shape is a bijective function $\Pi : \Omega \to S$ that maps points on the base domain Ω to points on the shape's surface. The parameterization of a triangle mesh $M = (\mathcal{V}, \mathcal{F})$ can itself be represented as a mesh $P = (\mathcal{U}, \mathcal{F})$ with $\mathcal{U} = \{\boldsymbol{u}_1, \dots, \boldsymbol{u}_m\}$. Note that M and P have the same number of points and identical triangulation. The point $\boldsymbol{u}_i$ is called the *parameter space coordinate* of $\boldsymbol{v}_i$, which means that $\Pi(\boldsymbol{u}_i) = \boldsymbol{v}_i$. In order to define the parameterization not only for the points in $\mathcal{U}$, but for any point of the base domain Ω, barycentric mapping is used. For every point $\boldsymbol{u} \in \Omega$, there is some triangle $\{i, j, k\} \in \mathcal{F}$ such that

$$\boldsymbol{u} = (1 - \nu_1 - \nu_2)\boldsymbol{u}_i + \nu_1 \boldsymbol{u}_j + \nu_2 \boldsymbol{u}_k, \tag{3.2}$$

where ν_1, ν_2 are barycentric coordinates, that is $\nu_1, \nu_2 \geq 0$ and $\nu_1 + \nu_2 \leq 1$. The point $\boldsymbol{v} = \Pi(\boldsymbol{u})$ is then given by the equation

$$\boldsymbol{v} = (1 - \nu_1 - \nu_2)\boldsymbol{v}_i + \nu_1 \boldsymbol{v}_j + \nu_2 \boldsymbol{v}_k. \tag{3.3}$$

A mesh parameterization algorithm computes an inverse mapping $\Pi^{-1}(\boldsymbol{v}_i) = \boldsymbol{u}_i$ for any mesh point $\boldsymbol{v} \in \mathcal{V}$. The inverse mapping is constrained such that the resulting parameterization mesh P is *valid*. This means that the whole surface of the base domain is covered with triangles, and that triangles do not fold or overlap. The latter constraint ensures that the mapping is bijective.

Parameter space coordinates $\boldsymbol{u}$ are 2D Cartesian coordinates for planar domains, and 3D Cartesian coordinates for the unit sphere. In the latter case, it is assumed that $\|\boldsymbol{u}\| = 1$. Because this property is not fulfilled when using the barycentric combination in Equation 3.2, the parameter space coordinates must be explicitly normalized to unit length. For the spherical domain, it is in some cases more convenient to use polar coordinates to represent points on the sphere. The polar coordinates of a spherical point $\boldsymbol{u}$ are denoted by $\boldsymbol{u}^\circ = (\theta, \varphi)$, where $\theta \in [0, \pi]$ is the colatitude and $\varphi \in [0, 2\pi)$ is the longitude.

In this chapter, registration of meshes is a reoccurring subproblem. Mesh registration algorithms, such as the ICP algorithm [BM92], compute their transformation usually based on closest point relations between point sets. For a point $\boldsymbol{v} \in \mathbb{R}^3$ and a point set $\mathcal{V} \subset \mathbb{R}^3$,

$$\mathrm{C}(\boldsymbol{v};\mathcal{V}) = \mathrm{argmin}_{\boldsymbol{v}' \in \mathcal{V}} \|\boldsymbol{v} - \boldsymbol{v}'\| \tag{3.4}$$

is the *nearest neighbor* in $\mathcal{V}$ to $\boldsymbol{v}$. Similarly, the set $\mathcal{C}_q(\boldsymbol{v};\mathcal{V}) \subset \mathcal{V}$ contains the q nearest neighbors to $\boldsymbol{v}$. Strictly speaking, neither the nearest neighbor nor the set of q nearest neighbors are always uniquely defined, but ties can be broken arbitrarily. (q-)Nearest neighbor search can be done efficiently when $\mathcal{V}$ is stored in a spatial data structure called kd-tree (see for example de Berg et al. [dBCvKO08], Chapter 5). Whenever a nearest neighbor search is described in this chapter, its implementation has been realized using kd-trees.

3.2. Establishing correspondence for shapes with spherical topology

This section discusses methods for establishing correspondence for shapes with spherical topology. It focuses on consistent spherical parameterization of shapes, in which a group of shapes is consistently mapped to the unit sphere. Additionally, it includes the description of a state-of-the-art groupwise optimization algorithm, which can be used to refine correspondences that have been initially provided by consistent parameterization.

Spherical parameterization is of high practical importance, because many organ shapes have spherical topology: Examples are the liver, the kidneys, the lungs, the ventricles of the heart, the spleen or the prostate. The unit sphere provides a natural parameter domain for these shapes: It is continuous, such that it is not necessary to cut the mesh in order to facilitate parameterization.

As discussed in Section 2.3.3.2, there are two potentially conflicting objectives when designing consistent parameterization methods: *Low distortion* of angles and areas and *consistency*. Low angle and area distortion simplifies the reconstruction of shapes from a parameter space: For low-distortion parameterizations, it suffices to use uniform sampling patterns on the parameter domain in order to sample landmark vectors that preserve the details of original input geometry well. Additionally, Kirschner et al. [KGWB11] have shown that low distortion makes the parameterizations more amenable to subsequent correspondence optimization. Consistency aims at mapping corresponding points on different shapes to the same points of the base domain. The use of consistent parameterizations facilitates faster convergence of groupwise optimization, and avoids that the optimization ‘gets stuck’ in poor local optima [DTT06].

Section 3.2.1 proposes an algorithm that constructs a spherical parameterization of a shape with low angle and area distortion. Based on this algorithm, two different methods for computing consistent spherical parameterization are devised: Alignment of spherical parameterizations (Section 3.2.2) and Parameter Space Propagation (Section 3.2.3). Although both methods can be used to construct high quality shape models on their own, they have been originally designed to provide a good starting point for groupwise optimization. To this end, a state-of-the-art groupwise optimization algorithm has been developed, which is described in Section 3.2.4. The question on how much the model quality can be further improved by this subsequent optimization is addressed in the evaluation in Chapter 4.

3.2.1. Spherical parameterization

In the following, an algorithm is described that computes a spherical parameterization $P = (\mathcal{U}, \mathcal{F})$ of a given mesh $M = (\mathcal{V}, \mathcal{F})$. The algorithm consists of four subsequent steps. In the first step, an initial, valid spherical parameterization of M is computed by solving two systems of linear equations. Once this initial mapping has been computed, it is modified in the following three steps in order to reduce distortion. In the second and the third step, reparameterization functions called *Sigmoid* and *CDF Reparameterization* are applied which reduce area distortion globally. In the fourth and final step, a stretch metric is minimized that accounts for both area and angular distortion on a local level.

3.2.1.1. Initial parameterization

The initial mapping of the mesh to the sphere is done using the method proposed by Brechbühler et al. [BGK95], which has been initially designed to parameterize voxel grids, but can be easily adapted to triangle meshes [SHM07].

The mesh is mapped to the sphere by solving two systems of linear equations, one for fixing the colatitude, and the other for fixing the longitude of spherical points. As boundary conditions for the linear equations, two distinct points must be determined, which are called *north* and *south pole*. For the same reason, a path called *date line* is calculated, which connects north and south pole.

North and south pole must not lie close together in order to avoid extreme distortions in the initial mapping. They are determined using a heuristic of Brechbühler et al. [BGK91]: It starts with selecting a random point, $\boldsymbol{v}_r \in \mathcal{V}$, of the mesh. A breadth first search [CLRS01] on G_M is used to compute the edge length of any node $\boldsymbol{v}_i \in \mathcal{V}$ to $\boldsymbol{v}_r$. The north pole $\boldsymbol{v}_{\text{north}}$ is drawn from the set of nodes with maximal edge distance to $\boldsymbol{v}_r$. The south pole is determined using a second breadth first search that starts in $\boldsymbol{v}_{\text{north}}$. Again, the south pole is defined to be a node with maximal edge distance to $\boldsymbol{v}_{\text{north}}$. The byproduct of the second breadth first search is the date line, which is the shortest path that connects north and south pole.

North and south pole are mapped to the sphere by defining $\boldsymbol{u}^{\circ}_{\text{north}} = (0,0)$ and $\boldsymbol{u}^{\circ}_{\text{south}} = (\pi,0)$. Moreover, the longitude of any point $\boldsymbol{v}_j$ on the date line is fixed by defining $\varphi_j = 0$. The remaining colatitude and longitude coordinates are determined by solving two systems of linear equations based on the graph Laplacian of G_M. In the first system, the fixed north and south pole provide boundary conditions for computing the colatitudes of all other nodes. Likewise, poles and date line provide boundary conditions when computing the longitude. Details for defining the linear systems can be found in the work of Brechbühler et al. [BGK95] or Schoening et al. [SHM07]. An example of the initial parameterization is shown in Figure 3.1.

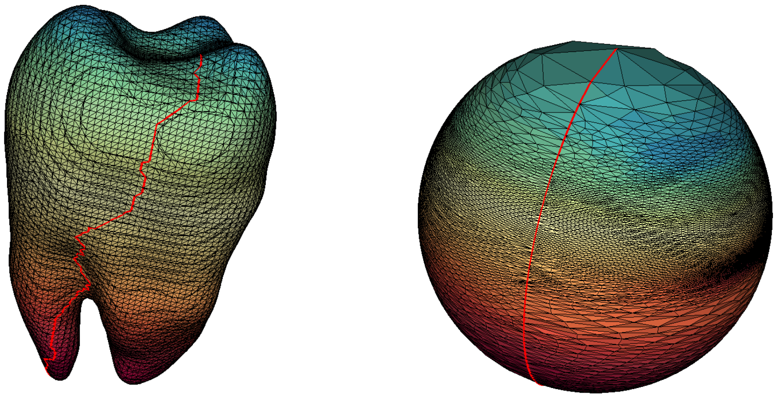


Figure 3.1.: Initial spherical parameterization of a tooth (approximately 12k vertices and 24k edges) using the method of Brechbühler et al. [BGK95]. The coloring encodes correspondence between mesh points and spherical coordinates. The red line is the date line connecting north and south pole.

3.2.1.2. Reducing distortions

The initial parameterization maps the majority of mesh points close to the 'equator' ($\theta = \frac{\pi}{2}$) and only few points close to the poles (see Figure 3.1). It is apparent that this parameterization has large angle and area distortion. For the parameterization of voxel grids, Brechbühler et al. [BGK95] reduce the distortion with a constrained nonlinear program. However, this nonlinear program assumes that all angles are perpendicular, which is true for voxel grids, but not for triangle meshes.

Most mesh parameterization algorithms that minimize distortion employ local optimization, where only one point is manipulated at a time (e.g. [PH03b, DMK03]). In these works, global reduction of distortion is achieved by using hierarchical mesh representations such as progressive meshes [Hop96]. In this work, an alternative approach is used: Reparameterization functions are employed for global reduction of area distortion. The idea of reparameterization has been successfully used for groupwise optimization. As explained in Section 2.3.3, reparameterization functions are diffeomorphic, which implies that they do not introduce triangle flips when they are applied to a valid parameterization. After the global reduction of area distortions using reparameterization, a standard local optimization scheme is used to refine the parameterization in the fourth and final step. The stretch metric of Degener et al. [DMK03] which is minimized takes both area and angle distortion into account.

Sigmoid Reparameterization The second step of the parameterization algorithm uses *Sigmoid Reparameterization* in order to reduce area distortion globally. This step is tailored to the properties of the initial parameterization. It spreads the points that initially tend to cluster around the equator more evenly on the sphere. Only the colatitude of the spherical points is affected by Sigmoid Reparameterization.

Sigmoid Reparameterization maps a spherical point (θ, φ) to the point $(f_{\text{sig}}(\theta), \varphi)$, where

$$f_{\text{sig}}(\theta) = \frac{\pi}{1 + \exp\left(-\alpha(\theta - \bar{\theta})\right)}. \tag{3.5}$$

Here, $\bar{\theta}$ is the inflection point of the sigmoid, which is set to the mean of the colatitude of all points of the initial parameterization. The parameter α controls the slope of the sigmoid, and is chosen such that the reparameterization minimizes the area distortion

$$\sum_{t \in \mathcal{F}} \left(\frac{A_M(t)}{\sum_{t' \in \mathcal{F}} A_M(t')} - \frac{A_P(t)}{\sum_{t' \in \mathcal{F}} A_P(t')} \right)^2, \tag{3.6}$$

where $A_M(t)$ and $A_P(t)$ denote the area of a triangle t on the mesh and the sphere, respectively. The optimal value for α is determined by numeric optimization using Brent's line search algorithm [PTVF92].

CDF Reparameterization. Although the Sigmoid Reparameterization affects only the colatitude of the spherical coordinates, a series of different reparameterizations can be made by randomly rotating the sphere each time before the sigmoid is applied. However, while the sigmoid function is particularly well suited for improving the initial parameterization with its characteristic point distribution, it does not help to reduce area distortion significantly in general. Instead of using a parametric function which only describes a very limited class of reparameterizations, it is desirable to reduce the area distortion with a diffeomorphic function with a shape that is *tailored to the current parameterization*. This leads to the idea of using cumulated density functions (CDFs) as reparameterization functions, a technique that is inspired by the concept of histogram equalization in image processing (see, for example, Gonzalez and Woods [GW08]).

CDF reparameterization is based on the following model: An infinitesimal small mass $\mu_{\boldsymbol{z}}$ is associated with each point $\boldsymbol{z}$ on the continuous surface of the shape such that the shape's total mass is uniformly distributed on the whole surface. A distortive mapping of the shape to the sphere compresses or stretches this mass on the spherical surface. The resulting mass distribution on a spherical region can be measured with a density function. Let $p(\theta)$ denote a one-dimensional density function that measures the density of spherical points with colatitude θ such that $p(\theta) \geq 0$ and $\int_{\theta=0}^{\pi} p(\theta) d\theta = 1$. The CDF induced by some parameterization algorithm is

$$F(\theta) = \int_{\{\boldsymbol{z} \mid \theta_{\boldsymbol{z}} \leq \theta\}} \mu_{\boldsymbol{z}} d\boldsymbol{z}. \tag{3.7}$$

In case of an area-preserving mapping, the uniform density distribution is the function

$$p_U(\theta) \mapsto \frac{1}{2} \sin(\theta), \tag{3.8}$$

because the circumference of the circle formed by points with colatitude θ is $2\pi \sin(\theta)$. The final function is obtained by dividing through 4π, which is the surface area of the unit sphere.

The corresponding CDF $F_U : [0,\pi] \to [0,1]$ of the uniform density $p_U(\theta)$ is

$$F_U(\theta) = \int_{\theta'=0}^{\theta} \frac{1}{2} \sin(\theta') d\theta' = \frac{1}{2}(1 - \cos(\theta)). \tag{3.9}$$

The idea of CDF Reparameterization is to transform the CDF F of the current parameterization to F_U, that is, to the CDF of the uniform density. In practice, the continuous surface of a shape has to be approximated by a discrete mesh with points and triangles. A weight

$$w(t) = \frac{A_M(t)}{\sum_{t' \in \mathcal{F}} A_M(t')} \tag{3.10}$$

is assigned to each triangle t which corresponds to the mass associated with it. Note that the weights are normalized such that they sum up to one. Furthermore, let $\bar{t}$ denote the center point of t. The CDF of the current parameterization is approximated by

$$F(\theta) \approx \sum_{t \in \mathcal{F}_\theta} w(t). \tag{3.11}$$

with $\mathcal{F}_\theta = \{t \mid \theta_{\bar{t}} \leq \theta\}$. Assume that the indexing of the triangles sorts them in ascending order of their colatitude, that is $\theta_{\bar{t}_i} \leq \theta_{\bar{t}_j}$ if $i < j$. In order to achieve an approximately uniform distribution of mass, $\bar{t}$ is mapped from (θ_i, φ_i) to (θ_i^*, φ_i), where

$$\theta_i^* = F_U^{-1}\left(\sum_{k=1}^{i} w(t_k)\right) = \operatorname{acos}\left(-2 \cdot \sum_{k=1}^{i} w(t_k) + 1\right). \tag{3.12}$$

In order to define a mapping for arbitrary points, the mapping $\theta_i \mapsto \theta_i^*$ can be used to define a piecewise linear function. Like the Sigmoid Reparameterization, the CDF Reparameterization affects only the colatitude coordinates. Therefore, the sphere is rotated with a randomly chosen rotation matrix using the method of Arvo [Arv92] before reparameterization. Usually, a few iterations of rotating the sphere and applying a CDF Reparameterization suffice to obtain a parameterization with small global area distortion.

Local stretch minimization. The methods proposed so far are well suited for reducing global area distortion. However, they do not necessarily account for local distortions, and ignore distortion of angles. In the fourth and final step of the algorithm, the parameterization is therefore refined by local minimization of the stretch metric of Degener et al. [DMK03], which accounts both for area and angle distortions. For each point $\boldsymbol{u}_i$ of the parameterization, Degener's energy is defined as the product of area and angle distortion of the triangles adjacent to $\boldsymbol{u}_i$:

$$E_{\text{Degener}}(\boldsymbol{u}_i) = \sum_{\{t \mid i \in t\}} E_{\text{angle}}(t) E_{\text{area}}(t)^{\vartheta}. \tag{3.13}$$

The exponent ϑ is used to balance the importance of area and angle distortion. Throughout this thesis, $\vartheta = 1$ is used.

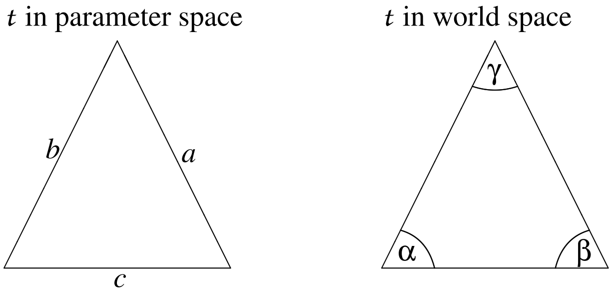


Figure 3.2.: Naming of angles and faces of a triangle t in parameter and world space.

The angle energy is defined by

$$E_{\text{angle}}(t) = \frac{\cot\alpha|a|^2 + \cot\beta|b|^2 + \cot\gamma|c|^2}{2A_P(t)} \tag{3.14}$$

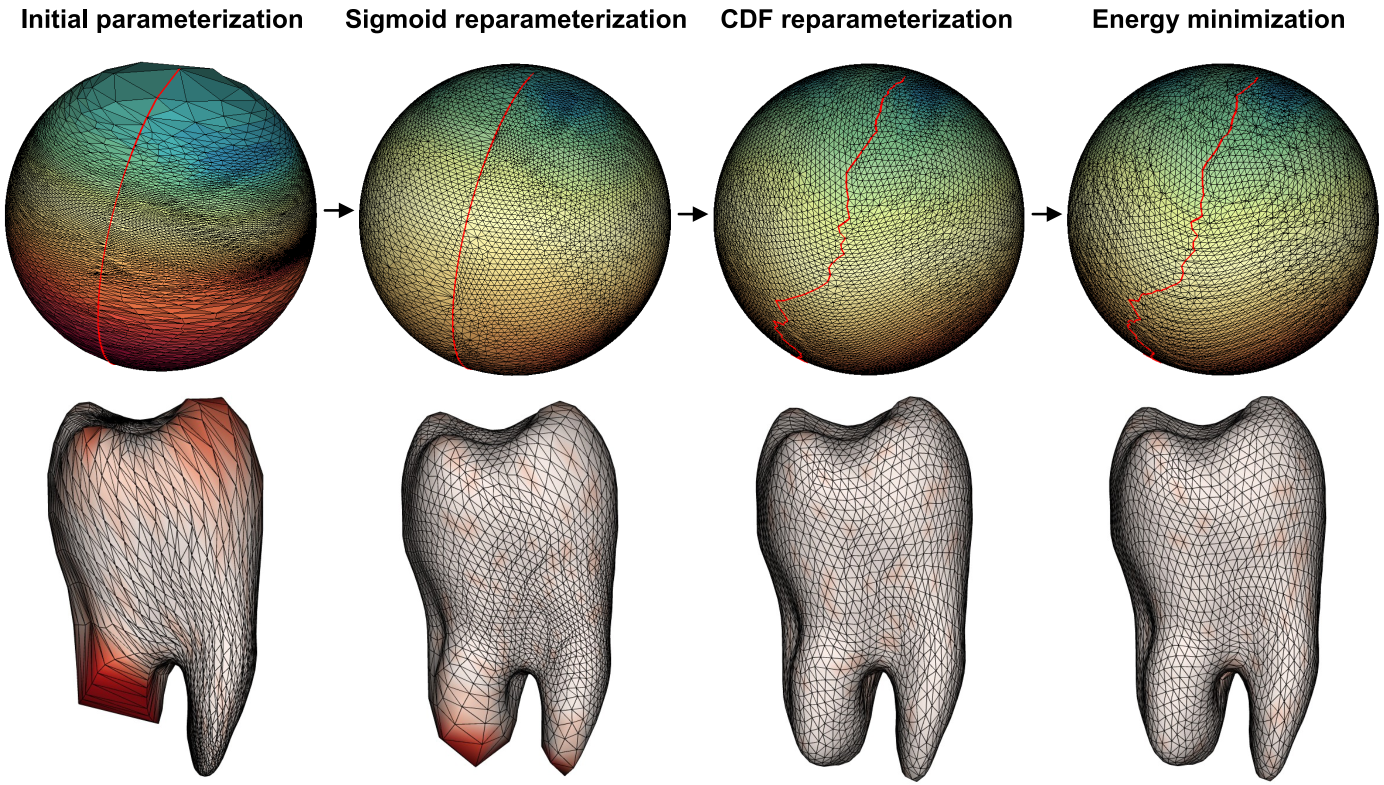
and the area energy by

$$E_{\text{area}}(t) = \frac{A_M(t)}{A_P(t)} + \frac{A_P(t)}{A_M(t)}, \tag{3.15}$$

where the naming of angles and faces shown in Figure 3.2 is used. The reader is referred to the publication of Degener et al. [DMK03] for a detailed derivation of the energy.

The energy is minimized by manipulating the coordinates of a single point $\boldsymbol{u}_i$ at a time. For this, the nodes are sorted in descending order of their energy, which means that nodes adjacent to triangles with high distortion are optimized first. The energy at a point is minimized using the Polak-Ribière conjugate gradient method [PTVF92]. The gradient of the energy at a point is computed using finite differences. Constraints are added to the optimization such that no triangle flip occurs.

Figure 3.3 visualizes the results of the four steps of the parameterization process of a tooth. The input mesh consists of approximately 12k points and 24k triangles. The figure shows both the parameterization after each of the four steps as well remeshes sampled after each step. The remeshes have been sampled using 2562 uniformly distributed sampling points connected by 5120 triangles, which corresponds to a reduction in size by approximately 80 % compared to the original mesh. The large distortion of the initial parameterization leads to a poor quality of the corresponding remesh that does not preserve all details of the input geometry. In the subsequent steps, the distortion becomes gradually smaller and the quality of the remeshes improves. The remeshes sampled after the third and fourth step of the algorithm approximate the original mesh with very small error, even though they have much less points. The triangles of the final remesh are slightly more regular than that after CDF Reparameterization, which shows that the final parameterization has indeed higher quality.



Initial parameterization

Sigmoid reparameterization

CDF reparameterization

Energy minimization

Figure 3.3.: Parameter spaces (first row) and corresponding remeshes (second row) after the four different steps of the spherical parameterization process. All remeshes have been constructed by uniform sampling in the parameter space with 2562 sampling points. On the remeshes, the error compared to the original mesh is visualized, where white means no error and red corresponds to a large error.

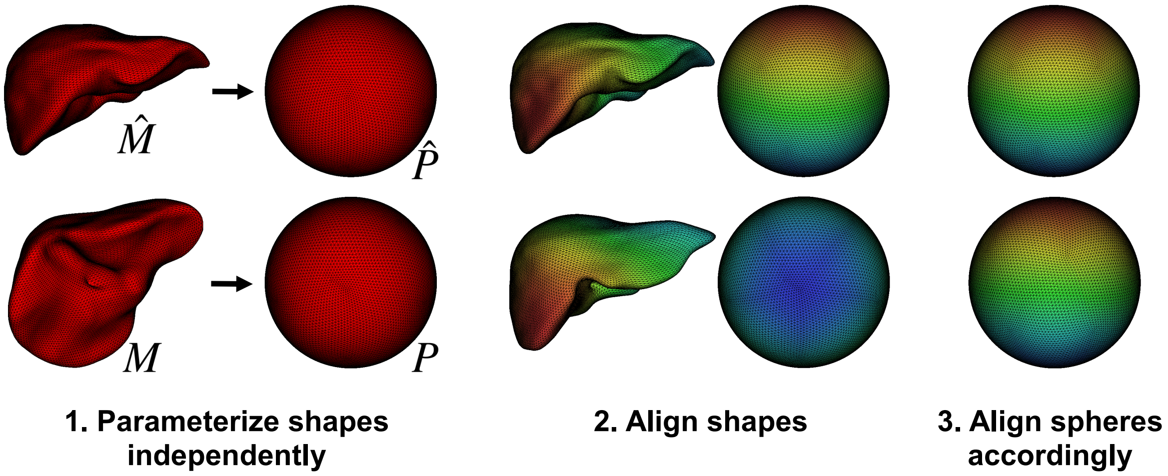


Figure 3.4.: Consistent alignment of spherical shape parameterization: Both shapes are independently mapped to the sphere. By aligning both meshes, a correspondence relation between points is derived, which is used to align the spheres.

3.2.2. Alignment of spherical shape parameterizations

When the parameterization method described in the previous section is applied to two training shapes independently, the resulting parameterizations are in general not consistent: The algorithm does not incorporate any means that ensure that corresponding features in both shapes are mapped to identical parameter space regions. A simple, but effective way to achieve consistency is to align the parameterizations afterwards, by keeping the first parameterization fixed and rotating the second such that they are optimally superimposed. The optimal rotation matrix used for this transformation is determined based on approximate point correspondences which are computed by registering the shapes. This idea, which is elaborated in more detail below, has been previously used by Fripp et al. [FBM*05] and Kirschner et al. [KGWB11]. In contrast to the aforementioned approaches, the method described here uses not an asymmetric, but a symmetric variant of the ICP algorithm [BM92].

The algorithm takes as input two meshes $M = (\mathcal{V},\mathcal{F})$ and $\hat{M} = (\hat{\mathcal{V}},\hat{\mathcal{F}})$ and computes consistent parameterizations $P = (\mathcal{U},\mathcal{F})$ and $\hat{P} = (\hat{\mathcal{U}},\hat{\mathcal{F}})$ of these meshes. It can be summarized as follows:

1. The algorithm described in Section 3.2.1 is used to create two low-distortion parameterizations P and $\hat{P}$.
2. M is aligned with $\hat{M}$ using a symmetric ICP. In the standard ICP, the optimal translation vector and rotation matrix are estimated in each iteration based on an asymmetric correspondence relation that relates every point $\boldsymbol{v} \in \mathcal{V}$ with its currently closest point $\mathrm{C}(\boldsymbol{v},\hat{\mathcal{V}})$. The symmetric ICP takes also the reverse direction into account, that is, it relates every point $\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}$ with its closest point $\mathrm{C}(\hat{\boldsymbol{v}},\mathcal{V})$. For each of the two closest point relations, an—in a least squares sense—optimal rotation matrix and translation vector is computed using the method of Horn [Hor87]. The final

transformation is computed by linear interpolation of the two translation vectors and spherical linear interpolation [Sho85] of the two rotation matrices.

3. After alignment, the two final closest point relations between $\mathcal{V}$ and $\hat{\mathcal{V}}$ are considered. These relations are transferred to the spherical domain, where they define relations between the spherical point sets $\mathcal{U}$ and $\hat{\mathcal{U}}$. For each relation, a rotation matrix that optimally superimposes the spherical points is computed using the method of Horn [Hor87]. Again, both rotation matrices are combined to a single rotation matrix using spherical linear interpolation.
4. The rotation matrix computed in the previous step is applied to all points $\boldsymbol{u} \in \mathcal{U}$ in order to superimpose P with $\hat{P}$.

Figure 3.4 shows an illustration of this algorithm.

The described algorithm aligns the parameterizations rigidly. This has the advantage that the transformation does not change areas or angles, such that the aligned spheres have identical distortion as the two independent, low-distortion parameterizations computed in the first step of the algorithm. The disadvantage of rigid alignment is that it greatly restricts the degree of consistency that can be achieved, because corresponding regions on different shapes may have different relative size.

3.2.3. Propagation of spherical shape parameterizations

Rigid alignment of two independently generated parameterizations, as described in the previous section, permits only a limited degree of consistency. This section proposes an alternative approach for constructing consistent spherical parameterizations, *Parameter Space Propagation*, which allows to achieve a higher degree of consistency at the costs of introducing slightly higher distortions. Instead of creating parameterizations independently and aligning them afterwards, Parameter Space Propagation uses a bootstrapping approach in which a shape parameterization P of M is constructed from an already existing parameterization $\hat{P}$ of a reference shape $\hat{M}$. The basic idea of the approach is that if M and $\hat{M}$ are similar, than the parameterizations P and $\hat{P}$ should be similar as well.

Parameter Space Propagation does not explicitly try to minimize global area distortions. This is because it is assumed that the parameterizations $\hat{P}$ and P are similar, which means that if $\hat{P}$ has low distortion, then the propagated parameterization P will also have low distortion. While this is certainly a heuristic assumption, it works very well on organ shapes, which is shown in the evaluation in Chapter 4. On a local level, however, where the shapes differ in detail, this argument no longer holds, and distortions of angles and areas occur. These distortion are explicitly handled by local stretch minimization.

The propagated parameterization often has some folded triangles, such that the mapping from the parameter domain to the mesh is not bijective. These foldings are caused by topologically inconsistent correspondences computed during registration. This happens because point set registration algorithms such as the ICP usually only consider the proximity of points, but ignore the triangulation. Parameter Space Propagation handles foldings by a correction strategy that identifies points that lie on folded triangles, deletes them from the parameterization, and reinserts them using interpolation.

As mentioned above, Parameter Space Propagation exploits similarity of shapes, which is algorithmically realized through mesh registration. This means that the results of the propagation process depends on the capabilities of the employed registration algorithm. In this work, the ICP algorithm is used for that purpose. It is likely that the results obtained by propagation can be further improved by using a more sophisticated algorithm, for example an algorithm that is also capable of nonrigid alignment. While the ICP considers only the nearest neighbor relation of points, a fuzzy, one-to-many correspondence relation is used during propagation. The weights that define this relation are not based on a theoretical framework, but were chosen empirically. If a stronger theoretical foundation is desired, registration algorithms such as the EM-ICP [GP06] can be used, whose probabilistic model inherently defines correspondence probabilities between pairs of points. These can be exploited for defining the propagation.

In the following, Parameter Space Propagation is explained in detail. It constructs a parameterization $P = (\mathcal{U}, \mathcal{F})$ of a mesh $M = (\mathcal{V}, \mathcal{F})$, given the parameterization $\hat{P} = (\hat{\mathcal{U}}, \hat{\mathcal{F}})$ of a reference mesh $\hat{M} = (\hat{\mathcal{V}}, \hat{\mathcal{F}})$. It is assumed that $\hat{P}$ has low distortion, which is achieved by constructing it using the algorithm described in Section 3.2.1. The propagation algorithm works as follows:

1. Align M with $\hat{M}$ in order to identify approximate point correspondences.
2. Based on the approximate point correspondences, construct P from $\hat{P}$.
3. Handle folded triangles of P.
4. Refine P by local minimization of area- and angle distortion.

These four steps are explained in more detailed below.

3.2.3.1. Shape Alignment

The goal of shape alignment is to identify approximate correspondences by superimposing M on $\hat{M}$. Initially, both meshes are centered, and M is anisotropically rescaled in order to account for size differences to $\hat{M}$. The three axes on which is scaled, as well as the corresponding scale factors, are determined by PCA of the mesh nodes of M and $\hat{M}$, respectively. More specifically, let $\boldsymbol{C} = \frac{1}{m} \sum_{\boldsymbol{v} \in \mathcal{V}} \boldsymbol{v}\boldsymbol{v}^T \in \mathbb{R}^3$ be the covariance matrix of the mesh nodes of M and $\hat{\boldsymbol{C}} = \frac{1}{\hat{m}} \sum_{\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}} \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^T \in \mathbb{R}^3$ be the covariance matrix of the mesh nodes of $\hat{M}$. Diagonalization yields matrix factorizations $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{\Gamma}\boldsymbol{B}^T$ and $\hat{\boldsymbol{C}} = \hat{\boldsymbol{B}}\hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{B}}^T$. The columns of the orthogonal matrices $\boldsymbol{B}$ and $\hat{\boldsymbol{B}}$ are the three principal axes of M and $\hat{M}$, respectively. The diagonal entries of the diagonal matrices $\boldsymbol{\Gamma}$ and $\hat{\boldsymbol{\Gamma}}$ contain the variances along the corresponding axes. Based on this information, the mesh $M' = (\mathcal{V}', \mathcal{F})$ is computed, which is an anisotropically scaled version of M. For $\boldsymbol{v} \in \mathcal{V}$, $\boldsymbol{v}' \in \mathcal{V}'$ is computed using the formula

$$\boldsymbol{v}' = \boldsymbol{B}\boldsymbol{\Gamma}'\boldsymbol{B}^T\boldsymbol{v}, \tag{3.16}$$

with $\boldsymbol{\Gamma}' = \operatorname{diag}(\hat{\gamma}_{11}/\gamma_{11}, \hat{\gamma}_{22}/\gamma_{22}, \hat{\gamma}_{33}/\gamma_{33})$. By construction, the meshes M' and $\hat{M}$ have the same extension along their principal axes.

After rescaling, M' is rigidly aligned with $\hat{M}$ using the ICP.

3.2.3.2. Propagation

From the aligned meshes M' and $\hat{M}$, a fuzzy, one-to-many correspondence relation is derived based on the proximity of points of $\mathcal{V}'$ and $\hat{\mathcal{V}}$. For all pairs $(\boldsymbol{v}'_i, \hat{\boldsymbol{v}}_j) \in \mathcal{V}' \times \hat{\mathcal{V}}$, a weight w_{ij} measures how strong $\boldsymbol{v}'_i$ and $\hat{\boldsymbol{v}}_j$ correspond. It is defined by

$$w_{ij} = \begin{cases} 1 & \text{if } \|\boldsymbol{v}'_i - \hat{\boldsymbol{v}}_j\| = 0 \\ \exp\left(-\frac{\|\boldsymbol{v}'_i - \hat{\boldsymbol{v}}_j\|^2}{2\sigma_i^2}\right) & \text{if } 0 < \|\boldsymbol{v}'_i - \hat{\boldsymbol{v}}_j\| \leq 4.5\sigma_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

The use of the exponential function ensures that the correspondence weight w_{ij} decreases quickly with increasing distance between $\boldsymbol{v}'_i$ and $\hat{\boldsymbol{v}}_j$. The scale factor σ_i used in the exponential function is set to the solution of the equation

$$\exp\left(-\frac{\|\boldsymbol{v}'_i - \mathrm{C}(\boldsymbol{v}'_i; \hat{\mathcal{V}})\|^2}{2\sigma_i^2}\right) = 0.75, \tag{3.18}$$

which implies

$$\sigma_i = \frac{1}{\sqrt{2(\ln(4) - \ln(3))}} \|\boldsymbol{v}'_i - \mathrm{C}(\boldsymbol{v}'_i; \hat{\mathcal{V}})\|. \tag{3.19}$$

The constant 0.75 has been chosen empirically. The general idea behind Equation 3.18 is that if the distance from $\boldsymbol{v}'_i$ to its nearest neighbor $\mathrm{C}(\boldsymbol{v}'_i; \hat{\mathcal{V}})$ is large, then there is a large uncertainty to which points $\boldsymbol{v}'_i$ corresponds. Accordingly, a larger scale factor σ_i is used, such that more points contribute to its parameter space coordinates $\boldsymbol{u}_i$.

For propagating the parameterization $\hat{P}$ to M, the parameter space coordinates $\boldsymbol{u}_i \in \mathcal{U}$ of a point $\boldsymbol{v}_i \in \mathcal{V}$ are computed as a weighted, normalized, sum of points $\hat{\boldsymbol{u}}_j \in \hat{\mathcal{U}}$:

$$\boldsymbol{u}_i = \frac{\sum_{j=1}^{\hat{m}} w_{ij} \hat{\boldsymbol{u}}_j}{\|\sum_{j=1}^{\hat{m}} w_{ij} \hat{\boldsymbol{u}}_j\|}. \tag{3.20}$$

The normalization makes sure that the propagated coordinates $\boldsymbol{u}_i$ are on the unit sphere.

Note that the weights are defined such that if $M = \hat{M}$, the propagation constructs the parameterization $P = \hat{P}$.

3.2.3.3. Correction

The propagation of parameter space coordinates described above depends only on the proximity of points in the aligned meshes. The correspondences derived by this are not necessarily topologically consistent with the mesh triangulation. This means that after propagation, some triangles are inverted or fold on the sphere. This problem is addressed by a correction step, which iteratively identifies problematic triangles on the parameter space. All points that belong to these triangles are deleted from the parameterization and afterwards reinserted using interpolation. The correction algorithm works as follows:

1. Identify inverted triangles, which are those triangles whose surface normal point to the inside of the sphere. If no inverted triangle exists, stop.
2. Determine the set $\tilde{\mathcal{U}} \subset \mathcal{U}$, which contains all points on the sphere that either belong to an inverted triangle, or belong to a triangle that intersects with an inverted triangle.
3. Partition $\tilde{\mathcal{U}}$ into several connected clusters using a connected component analysis (see, e.g., Cormen et al. [CLRS01]) on the subgraph of G_M induced by $\tilde{\mathcal{U}}$.
4. Reinsert each cluster to the sphere. This is done as follows:
   a) Identify the boundary points of the cluster, which are points that are adjacent to at least one cluster point, but do not belong to the cluster themselves.
   b) Compute the spherical mean of the boundary points using the algorithm of Buss and Fillmore [BF01].
   c) Map the boundary points to the tangent plane of their spherical mean using orthographic projection.
   d) Map all cluster points to the plane using Tutte parameterization [Tut63]. The boundary points provide the necessary boundary conditions.
   e) Use the inverse orthographic projection to map the cluster nodes back to the sphere.
5. Go to 1.

There is no guarantee that the correction algorithm converges. To ensure termination, it is stopped when a maximal number of iterations is exceeded. The algorithm assumes that all boundary points lie on the hemisphere which is defined by their spherical mean, because otherwise, the orthographic projection is ill-defined. Although these theoretical restrictions exist, the algorithm works well in practice and usually terminates after a few iterations.

3.2.3.4. Refinement

In the final step of Parameter Space Propagation, the parameterization P is refined by minimizing local angle and area distortion. To this end, Degener's energy [DMK03] is locally minimized, as it was done in Section 3.2.1.2. While Degener's energy does not take consistency into account, the minimization approach is localized, such that it does not destroy the global consistency of the parameterization.

3.2.4. Groupwise optimization

Consistent parameterization inherently defines a correspondences between meshes: Points with the same parameter space coordinates correspond to each other. This correspondence can be refined by groupwise optimization which modifies the correspondences by modifying the parameterizations, that is, by reparameterization.

This section describes a state-of-the-art groupwise optimization algorithm, whose implementation follows in most parts the description and recommendations of Davies et al. [DTT08b, DTCT10]. The

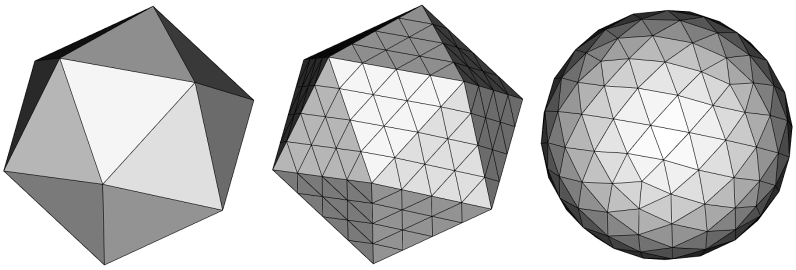


Figure 3.5.: An approximately uniform sampling pattern for the unit sphere can be generated by subdividing the faces of a regular unit icosahedron and projecting the interpolated points to sphere.

optimization minimizes the model-based objective function F_{MDL} (Equation 2.20). Reparameterization is implemented by moving sampling points on the sphere, rather than modifying the parameterizations itself. An illustration of this optimization concept is illustrated in Figure 2.6 on page 36. If groupwise optimization is not desired, shape models can also be directly constructed from the consistent shape parameterization. In this case, the algorithm stops after initialization (Section 3.2.4.1).

The optimization algorithm described here can also be used to optimize shapes of genus 1. In that case, the components that depend on the parameter domain must be replaced: the input parameterizations, the sampling pattern and the reparameterization function. Details for the implementation of these components for genus 1 optimization are given in Section 3.3.3.

3.2.4.1. Initialization

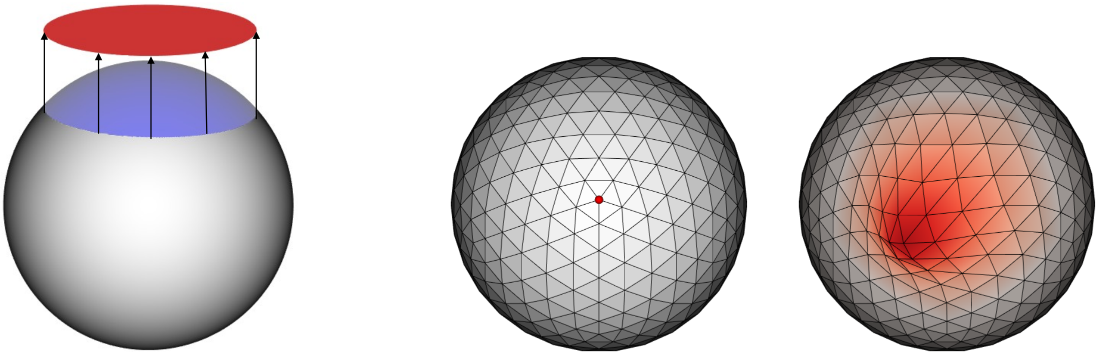
The input of the algorithm is a set of spherical shape parameterizations, which can be generated for example using the algorithms presented in Section 3.2.2 or Section 3.2.3.

For each shape, a set of n sampling points is created, which are used to reconstruct landmark vectors from the parameterizations. The landmark vectors are used to calculate the objective function. Figure 3.5 shows how an approximately uniform sampling pattern can be constructed: The faces of a regular unit icosahedron are uniformly subdivided into several triangles. The interpolated points are projected to the sphere by normalizing the coordinates.

Alternatively, an adaptive sampling pattern that accounts for distortions in the parameterization or curvature in the meshes can be generated using the method presented in Section 3.4.

3.2.4.2. Optimization

In order to optimize the objective function, an iterative, stochastic optimization scheme is used: In each iteration, a shape is drawn uniformly at random from the set of training shapes, and its sampling points are optimized by reparameterization. The reparameterization is steered by hyperparameters, which


(a) Orthographic projection

(b) Spherical CPS Reparameterization

Figure 3.6.: (a): Orthographic projection of a spherical cap (light blue) to the unit circle. (b): Effect of a CPS reparameterization: The control point, marked with a red point in the left sphere, is moved to the south west, and neighboring points are moved accordingly. The amount of deformation of the mapping is illustrated with a color mapping, where dark red corresponds to large deformation.

are chosen stochastically, and optimization parameters, which are explicitly optimized using gradient descent. The gradient is estimated numerically using finite differences. As recommended by Davies et al [DTT08b], the parameterization of one of the shapes is kept fixed during optimization in order to constrain the optimization process. The algorithm stops either when a maximum number of iterations is exceeded, or the algorithm converges. Convergence is reached when the objective function does not decrease by more than a predefined threshold within s iterations.

3.2.4.3. Reparameterization

Reparameterization of sampling points on the sphere is realized using Clamped Plate Splines (CPS), which has been proposed by Davies et al. [DTT06, DTT08b]. CPS reparameterization was chosen because its effects are localized and because it allows for coarse-to-fine optimization.

The reparameterization works as follows: First, a spherical cap is mapped to the unit circle using orthographic projection as shown in Figure 3.6(a). The cap's pole is mapped to the circle's center, and the boundaries of the cap are mapped to the circle's boundary. Next, the circle's center is moved to some other point on the circle, and all other points within the circle are moved accordingly using the Clamped Plate Spline that is defined by the movement of the center point. Finally, the modified circle can be backprojected to the sphere using inverse orthographic projection. Figure 3.6(b) illustrates the effect of a CPS reparameterization.

CPS reparameterization has two hyperparameters, the *center* and the *radius*, which control the size of the spherical cap that is modified. In each iteration of the optimization process, a set of non-overlapping spherical caps that are uniformly distributed on the sphere is selected using the icosahedron subdivision

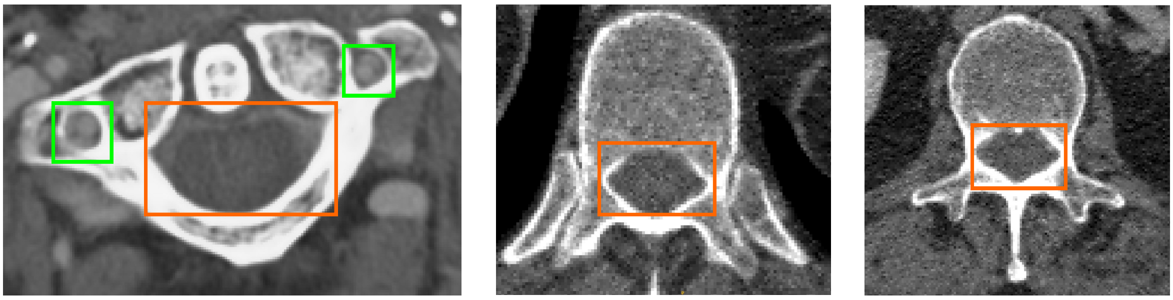


Figure 3.7.: From left to right: Cervical, thoracic and lumbar vertebra. The orange box marks the vertebral foramen which encloses the spinal cord. Cervical vertebrae have two additional foramina, here marked by green boxes.

strategy. Because the effect of the reparameterization is strongest at the caps, the caps are rotated with a randomly chosen rotation matrix such that all regions of the parameter space are equally affected by the optimization. The size of the caps is chosen such that the caps do not overlap. Each cap is then individually optimized. The optimization parameters are the coordinates of the points to which the circle's center is mapped.

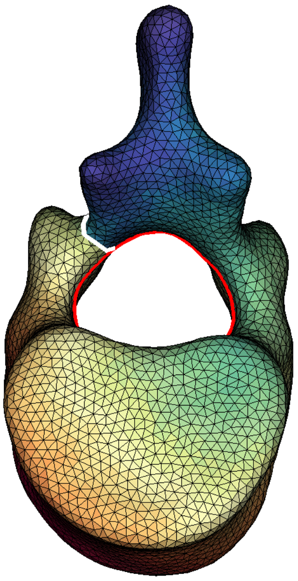
In coarse-to-fine optimization, the number of spherical caps is increased, while their size is decreased.

3.3. Establishing correspondence for shapes with toroidal topology

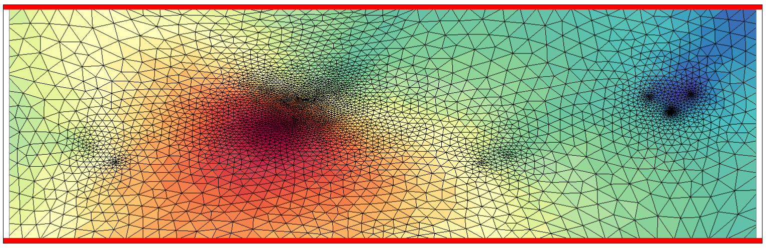
The algorithms discussed in Section 3.2 require that the training shapes can be mapped to the unit sphere. Although this is possible for most soft tissue organs, some bone structures have holes and therefore non-spherical topology. To these structures, spherical parameterization is not applicable.

An example for bone structures with holes are vertebrae. Figure 3.7 shows the three vertebrae types that can be found in the human body: cervical, thoracic and lumbar vertebrae. Thoracic and lumbar vertebrae are of genus 1 because they have exactly one hole: The vertebral foramen which encloses the spinal cord. Cervical vertebrae are of genus 3 due to two additional foramina, which are situated in the transverse processes and enclose blood vessels.

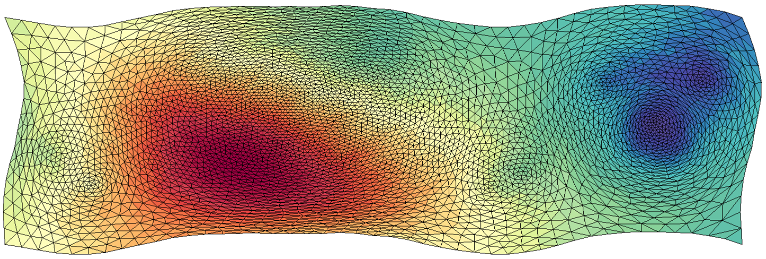
This section describes parameterization-based methods to establish correspondence for shapes of toroidal topology, which can be used, for example, to construct shape models of thoracic or lumbar vertebrae. The presented approach is very similar to that used for spherical parameterization, but it works instead on a planar parameter domain: Initially, a low distortion mapping of a single shape, the template, is computed (Section 3.3.1). All other shapes are parameterized in such a way that the resulting mappings have both low distortion and are consistent with respect to the template parameterization (Section 3.3.2). Shape vectors can be directly sampled from the consistent parameterization



(a) Thoracic vertebra (Th12)



(b) Initial parameterization



(c) Optimized parameterization

Figure 3.8.: Mapping of a thoracic vertebra (Th12) to the plane. (a): The original vertebra mesh with its system of shortest loops (red and white curve). (b): In the initial parameterization, the red loop is mapped to the horizontal boundaries, and the white loop to the vertical boundaries of the rectangular parameter space. (c): The final, low-distortion parameter space.

by sampling in the plane. Alternatively, the correspondence can be further optimized using groupwise optimization (Section 3.3.3).

The genus 1 parameterization algorithm has been devised by Meike Becker in her Master's thesis [Bec10] and has been extended to consistent parameterization by Becker et al. [BKFW11]. The latter approach has been further extended in this thesis by a new correspondence term, which is in contrast to the former symmetric. This correspondence term as well as groupwise optimization of shapes of genus 1 is exclusively my work.

3.3.1. Planar parameterization of shapes of genus 1

A rectangular planar domain is chosen as parameter space for shapes of genus 1, represented by a triangle mesh $M = (\mathcal{V}, \mathcal{F})$. In contrast to other planar parameterization methods, the parameter space is made continuous by dividing the whole plane into equivalence classes. For example, the unit square defines the following set of equivalence classes on the plane:

$$\left\{ \{\boldsymbol{u}' \mid u'_1 = n_1 + u_1 \text{ and } u'_2 = n_2 + u_2 \text{ for } n_1, n_2 \in \mathbb{N}\} \;\middle|\; \boldsymbol{u} \in [0,1)^2 \right\}. \tag{3.21}$$

Points that belong to the same equivalence class parameterize the same mesh point. In particular, the horizontal and vertical boundaries of the domain correspond to each other, as visualized in Figure 3.8(b).

In order to map M to the plane, it is cut along two loops, which are mapped to the boundaries of the rectangular parameter domain. The aspect ratio of the rectangle is chosen proportionally to the quotient of the lengths of these loops. The algorithm of Erickson and Whittlesey [EW05] is used to compute a system of shortest loops on the graph G_M, such that the two loops meet at a common point, the *base point*. Here, the length of a loop $(j_0, \dots, j_k)$, $j_0 = j_k$, is defined as

$$\sum_{i=1}^{k} \|\boldsymbol{v}_{j_{i-1}} - \boldsymbol{v}_{j_i}\|. \tag{3.22}$$

It can be shown that the algorithm of Erickson and Whittlesey [EW05] always finds exactly two loops for meshes that are of genus 1.

Figure 3.8 illustrates how the mesh points are mapped to the boundaries of the parameter domain. The longer loop, shown in red, is mapped to the horizontal boundaries, and the shorter loop, shown in white, to the vertical boundaries. From an implementational point of view, all points on the loops are represented by two points in parameter space, and the base point is actually represented by four points, which are the corners of the parameter space. However, mathematically, all representatives of a mesh point are mapped to the same equivalence class in parameter space. After having fixed the positions of points on the loops, the remaining points are embedded into the plane using Tutte parameterization [Tut63].

The resulting parameterization usually has large area and angle distortion. The distortion is reduced by minimizing the energy functional of Degener et al. [DMK03], which has already been introduced in Section 3.2.1.2 when discussing spherical parameterization. Due to the continuity of the parameter space, the loop points can leave the parameter space boundaries during optimization. When moving one of the boundary points, it is ensured that the corresponding point on the opposite boundary is moved as well such that both points remain in the same equivalence class. The effect of optimization is illustrated in Figure 3.8: In the initial parameterization, shown in Figure 3.8(b), all loop points lie on the boundaries of the rectangular parameter space. After optimization, most loop points have left the boundaries, which can be seen in Figure 3.8(c).

3.3.2. Consistent parameterization of shapes of genus 1

The goal of consistent parameterization is to generate a parameterization $P = (\mathcal{U}, \mathcal{F})$ of the mesh $M = (\mathcal{V}, \mathcal{F})$ such that P is consistent with an already existing parameterization $\hat{P} = (\hat{\mathcal{U}}, \hat{\mathcal{F}})$ of shape $\hat{M} = (\hat{\mathcal{V}}, \hat{\mathcal{F}})$. As it was done for spherical parameterizations, the ICP [BM92] is integrated into the parameterization process in order to achieve consistency.

Using the symmetric ICP described in Section 3.2.2, M is rigidly aligned with $\hat{M}$. Then, a system of loops on M is determined that is consistent with the loops on $\hat{M}$: The base point of M is chosen to be the point in $\mathcal{V}$ which is closest to the base point of $\hat{M}$. Again, a system of loops for the selected base point is computed using the algorithm of Erickson and Whittlesey [EW05]. In contrast to the parameterization of a single shape described in the previous section, the length of an edge $e = \{j,k\}$, is not defined as the Euclidean distance between $\boldsymbol{v}_j, \boldsymbol{v}_k \in \mathcal{V}$, but as follows:

$$c(e) = \min_{\hat{\boldsymbol{v}} \in \hat{\mathcal{L}}} \|\boldsymbol{v}_j - \hat{\boldsymbol{v}}\| + \min_{\hat{\boldsymbol{v}} \in \hat{\mathcal{L}}} \|\boldsymbol{v}_k - \hat{\boldsymbol{v}}\|. \tag{3.23}$$

Here, $\hat{\mathcal{L}} \subset \hat{\mathcal{V}}$ is the set of points that lie on the loops of $\hat{M}$. The cost function penalizes deviations from $\hat{\mathcal{L}}$ and thus ensures that the loops on M are consistent with the loops $\hat{M}$.

After finding the loops, they are mapped to the parameter space boundaries and Tutte parameterization [Tut63] is used to compute an initial parameterization of M. This parameterization is optimized by minimizing the following energy:

$$E(P;\hat{P}) = \sum_{\boldsymbol{u} \in \mathcal{U}} \left(E_{\text{consistent}}(\boldsymbol{u};\hat{P}) + \zeta \cdot E_{\text{degener}}(\boldsymbol{u})\right). \tag{3.24}$$

Here, $E_{\text{degener}}(\boldsymbol{u})$ is Degener's energy [DMK03] at point $\boldsymbol{u}$, $E_{\text{consistent}}(\boldsymbol{u})$ is a consistency term, and the free parameter ζ balances both energies. Thus, minimizing the energy corresponds to seeking a compromise between low distortion and consistency with the template parameterization $\hat{\mathcal{P}}$. The consistency term $E_{\text{consistent}}(\boldsymbol{u})$ is described in the following.

3.3.2.1. Consistency term

The consistency term enforces that points that lie nearby in the aligned meshes are also mapped to similar regions of the parameter space. A sparse correspondence matrix $\boldsymbol{A} \in \mathbb{R}^{m \times \hat{m}}$ with entries a_{ij} is constructed by q-nearest neighbor search from $\mathcal{V}$ to $\hat{\mathcal{V}}$ and vice versa. The entries a_{ij} define the degree of correspondence between the points $\boldsymbol{v}_i \in \mathcal{V}$ and $\hat{\boldsymbol{v}}_j \in \hat{\mathcal{V}}$, where a higher a_{ij} encodes a higher degree of correspondence.

Using the a_{ij}, the consistency term for a parameter space point $\boldsymbol{u}_i \in \mathcal{U}$ is defined as

$$E_{\text{consistent}}(\boldsymbol{u}_i;\hat{P}) = \sum_{j=1}^{\hat{m}} a_{ij} \|\boldsymbol{u}_i - \hat{\boldsymbol{u}}_j\|^2. \tag{3.25}$$

In order to define appropriate entries a_{ij} for the correspondence matrix, the following three requirements are considered: Firstly, symmetry requires that both surfaces have the same influence on the correspondence term, even if they have different number of points. Secondly, points that lie on the same surface should have the same overall weight on the correspondence term. Thirdly, the entries a_{ij} should sum up to m. This simplifies balancing the consistency term with Degener's energy.

These requirements are realized using the point weights defined in the following. For $\boldsymbol{v}_i \in \mathcal{V}$, let

$$\mathcal{C}_q^{-1}(\boldsymbol{v}_i;\hat{\mathcal{V}}) = \{\hat{\boldsymbol{v}}_j \in \hat{\mathcal{V}} \mid \boldsymbol{v}_i \in \mathcal{C}_q(\hat{\boldsymbol{v}}_j;\mathcal{V})\} \subset \hat{\mathcal{V}} \tag{3.26}$$

be the set of points to which $\boldsymbol{v}_i$ is a q-nearest neighbor. The weight w_i of a point $\boldsymbol{v}_i \in \mathcal{V}$ is defined by

$$w_i = \frac{1}{q + |\mathcal{C}_q^{-1}(\boldsymbol{v}_i; \hat{\mathcal{V}})|} \tag{3.27}$$

and the weight of a point $\hat{\boldsymbol{v}}_j \in \hat{\mathcal{V}}$ by

$$\hat{w}_j = \frac{m}{\hat{m}(q + |\mathcal{C}_q^{-1}(\hat{\boldsymbol{v}}_j; \mathcal{V})|)}. \tag{3.28}$$

Finally, the entries of the similarity term are defined by

$$a_{ij} = \begin{cases} w_i + \hat{w}_j & \text{if } \boldsymbol{v}_i \in \mathcal{C}_q(\hat{\boldsymbol{v}}_j; \mathcal{V}) \text{ and } \hat{\boldsymbol{v}}_j \in \mathcal{C}_q(\boldsymbol{v}_i; \hat{\mathcal{V}}) \\ \frac{1}{2}(w_i + \hat{w}_j) & \text{if } \boldsymbol{v}_i \in \mathcal{C}_q(\hat{\boldsymbol{v}}_j; \mathcal{V}) \text{ xor } \hat{\boldsymbol{v}}_j \in \mathcal{C}_q(\boldsymbol{v}_i; \hat{\mathcal{V}}) \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}$$

The following computations show that the definition of the matrix entries a_{ij} fulfill the requirements mentioned above:

$$\begin{aligned} \sum_{i=1}^{m} \sum_{j=1}^{\hat{m}} a_{ij} &= \sum_{i=1}^{m} \frac{1}{2}(q + |\mathcal{C}_q^{-1}(\boldsymbol{v}_i; \hat{\mathcal{V}})|) w_i + \sum_{j=1}^{\hat{m}} \frac{1}{2}(q + |\mathcal{C}_q^{-1}(\hat{\boldsymbol{v}}_j; \mathcal{V})|) \hat{w}_j && (3.30) \\ &= \sum_{i=1}^{m} \frac{1}{2} + \sum_{j=1}^{\hat{m}} \frac{1}{2} \frac{m}{\hat{m}} && (3.31) \\ &= m. && (3.32) \end{aligned}$$

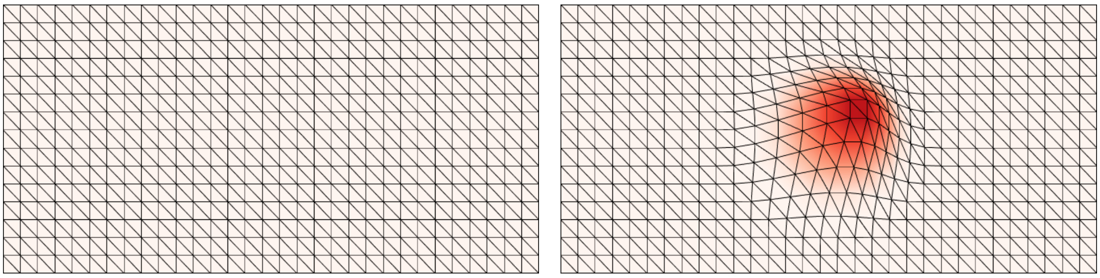
Equation 3.30 splits the sum into sums of w_i and $\hat{w}_j$. The weight $\frac{1}{2} w_i$ is added $(q + |\mathcal{C}_q^{-1}(\boldsymbol{v}_i; \hat{\mathcal{V}})|)$ times: for each of the q nearest neighbors of $\boldsymbol{v}_i \in \mathcal{V}$ as well as for each point $\hat{\boldsymbol{v}}_j$ to which $\boldsymbol{v}_i$ is nearest neighbor. A similar argument explains the summands within the second sum.

Equation 3.31 follows from the definition of w_i and $\hat{w}_j$. It can be directly seen that all points on the same surface contribute the same weight to the consistency term, and that both surfaces have a total weight of $\frac{1}{2} m$.

It is possible to incorporate additional features computed at mesh points into the definition of the matrix entries a_{ij}, such as for example curvature measures or the angle between surface normals. In the implementation made for this thesis, it is possible to enforce that $a_{ij} = 0$ if the angle between the normals of $\boldsymbol{v}_i \in \mathcal{V}$ and $\hat{\boldsymbol{v}}_j \in \hat{\mathcal{V}}$ exceeds a certain threshold. During q-nearest neighbor search, points are simply ignored when their normals are incompatible with the normal of the query point.

3.3.3. Groupwise optimization

The correspondences obtained with consistent parameterization can be further refined by groupwise optimization. This can in principle be done with the optimization algorithm described in Section 3.2.4, but the components assuming that the parameter space is the unit sphere must be replaced in order to handle a rectangular parameter space instead: This means that different methods for generating an initial sampling pattern and for reparameterization are necessary.



(a) Sampling pattern for rectangular parameter space (b) CPS reparameterization on the rectangle

Figure 3.9.: (a): A uniform sampling pattern for the planar parameter space. This pattern can be reparameterized using CPS reparameterization. (b): A CPS reparameterization which moves the center of gravity of the parameter space to the top-right. Points in the vicinity are moved accordingly. The amount of deformation of the mapping is illustrated with a color mapping, where dark red corresponds to large deformation.

3.3.4. Sampling in rectangular parameter spaces

Figure 3.9(a) shows a uniform sampling grid for a rectangular parameter space. The grid is computed by subdividing the edges of a rectangle that covers the whole parameter space. Because the rectangular parameter space is continuous for shapes of genus 1, the points on opposite borders correspond to each other. Therefore, sampling points on the upper and on the right border are removed and their neighbors are instead connected to the corresponding points on the opposite border.

As an alternative to the uniform grid, adaptive sampling patterns can be constructed using the method described in Section 3.4.

3.3.4.1. Reparameterization

Similar to the spherical parameter space, Clamped Plate Splines are used for reparameterization during correspondence optimization. Initially, a circular subregion of the parameter space is mapped to the unit circle. The position of the points on the unit circle are modified by displacing the circle's center point. This displacement defines a smooth Clamped Plate Spline (CPS) transformation which is used to move all other points on the circle. Finally, the points on the modified unit circle are mapped back to the parameter space. An example of such a CPS reparameterization on the rectangle is illustrated in Figure 3.9(b).

Four parameters must be chosen in order to define the transformation: the center of the circular subregion, its radius and the movement of the center point in x and y direction. The latter two parameters are explicitly optimized, while the center is set to a point on the rectangle that is chosen uniformly at random. The radius is kept fixed during optimization. It is possible to use coarse-to-fine optimization, by starting with a relatively large radius, and gradually reducing it in finer resolution levels.

Note that the circular region of the parameter space defined by center and radius may intersect with the boundaries of the parameter space. In that case, points lying on the opposite side of the parameter space are affected by the reparameterization as well in order to enforce continuity of the transformation. Sampling points that are pushed over the boundaries of the parameter space are reinserted on the opposite side.

3.4. Adaptive sampling in parameter space

In Section 3.2.4.1 and Section 3.3.4, *uniform* sampling patterns have been used in order to sample landmark vectors from spherical or planar parameter spaces. This section presents a generic method for *adaptive* sampling of landmark vectors.

Adaptive sampling methods sample certain regions of the parameter space with higher densities than others. They can be used, for example, to compensate for area distortion in the parameter spaces [HWM07]: By choosing the number of sampling points for a local parameter space region proportionally to the fraction of the surface area that is mapped to the region, it can be ensured that points are approximately uniformly distributed on the remeshed surface. Another application is curvature adaptive sampling: if more sampling points are used in regions of higher curvature, the characteristic details of the shape can be preserved even if the number of landmarks is relatively small. Using only few landmarks is desirable because it decreases the computation time, and reduces the dimensionality of the learning problem.

The method presented here is applicable to many different parameter domains: It only requires that a method for generating an approximately uniform sampling pattern for the domain is available. Moreover, many different criteria can be used for controlling the sampling density, such as distortion in the parameter space, curvature, or something completely different. Whichever criterium is used, it must be representable by a *density field* which associates a numerical value with every sampling point that represents a local region in the parameter space.

The method for adaptive sampling of landmark vectors from a set of parameterizations with the same base domain works as follows:

1. A dense, uniform sampling pattern is used to sample dense landmark vectors from each parameterizations, and to create a density field for the parameterization.
2. The landmark vectors are averaged to a dense mean mesh M_{dense}. Note that the uniform sampling pattern used in 1. is a parameterization P_{dense} of M_{dense}.
3. The density fields of all parameterizations are combined to a density field of P_{dense}.
4. The remeshing algorithm of Fuhrmann et al. [FAKG10] is used to create a sparsely and adaptively sampled mean shape M_{sparse}. Each point of M_{sparse} lies on some triangle of M_{dense}, and therefore,

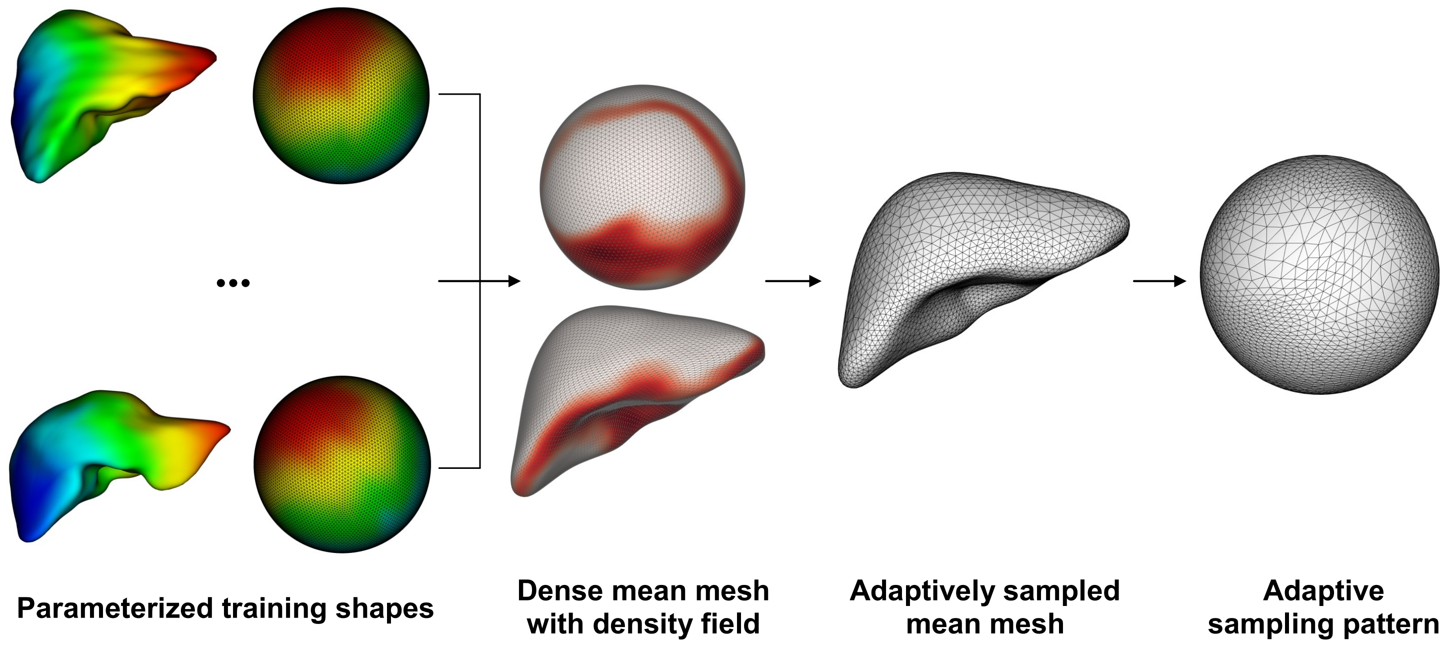


Figure 3.10.: Illustration of the computation of an adaptive sampling pattern for the unit sphere. A dense mean mesh is sampled from the parameterizations. Similarly, a density field is sampled for each node of the mean mesh. In this case, the density field encodes curvature at the mesh points, and is visualized using color coding, where white means low curvature and red means high curvature. The mean mesh is adaptively resampled, and the corresponding parameterization is interpolated. This parameterization constitutes the adaptive sampling pattern.

each point has a parameter space coordinates in P_{dense}. These parameter space points define a sparse parameterization P_{sparse} of M_{sparse}.

5. P_{sparse} is used as adaptive sampling pattern to sample the final landmark vectors from each parameter space.

An illustration of this algorithm is shown in Figure 3.10.

In this work, the density field is constructed by computing the absolute mean curvature of the mesh points of the dense landmark vectors. Dyn et al. [DHKL01] give formulas for calculating the mean absolute curvature on discrete meshes. The density fields are combined by averaging, which is especially appropriate when corresponding regions of the shapes have similar curvature. For shapes with high curvature variation in corresponding regions, as for example in data extracted from a time series of the moving heart, taking the maximal density per point is an alternative choice.

3.5. Establishing correspondence for shapes with arbitrary topology

The parameterization-based methods presented in Section 3.2 and Section 3.3 are restricted to shapes classes with either spherical or toroidal topology. In this section, a method based on nonrigid mesh registration is presented which is independent of the topology of the mesh. It can be used, for example, for organs with complex topology such as the semicircular canals shown in Figure 3.11. Furthermore, because the algorithm employs a new multi-scale optimization strategy, it is capable of computing smooth deformations even in case of large shape differences.

The nonrigid registration algorithm presented here computes a smooth displacement field that maps points of one input mesh to the surface of a second input mesh. This strategy is also used in some other nonrigid mesh registration algorithms such as the nonrigid EM-ICP [CP10] and the Coherent Point Drift [MS10]. Like the nonrigid EM-ICP, a symmetric similarity measure is used. This symmetry is extremely important for the correspondence problem, as the sought correspondence relation can be understood as a bijective function. However, the similarity measure is different to that of the EM-ICP: The objective of the algorithm presented here is to explicitly minimize the root mean square difference between the surfaces. The second difference to existing approaches is the employed multi-scale optimization strategy which allows for robust registration even in case of high shape variability.

The nonrigid registration algorithm is detailed in Section 3.5.1. The multiscale optimization approach employed for computing the registration is described in Section 3.5.2. Finally, Section 3.5.3 explains how the algorithm is used to construct corresponding landmark vectors.

3.5.1. Pairwise nonrigid registration

For modeling the objective function, the input shapes $\mathcal{S}$ and $\hat{\mathcal{S}}$ are first regarded as continuous surfaces. The goal of the registration is to deform $\mathcal{S}$ in such a way that it exactly matches $\hat{\mathcal{S}}$. That is, a function

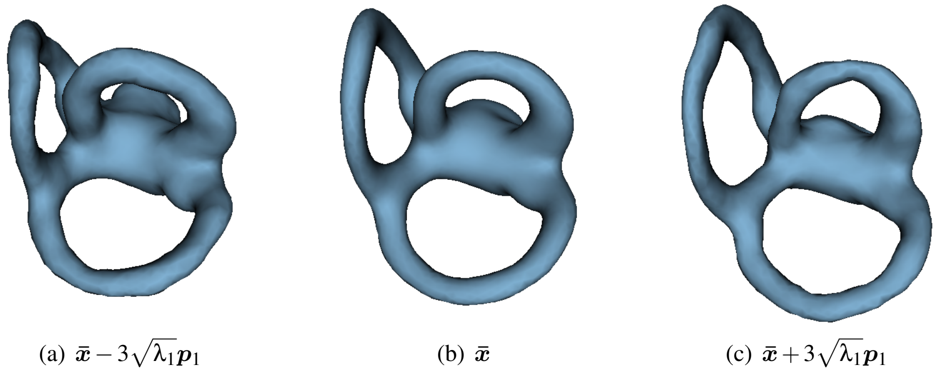

(a) $\bar{\boldsymbol{x}} - 3\sqrt{\lambda_1}\boldsymbol{p}_1$ (b) $\bar{\boldsymbol{x}}$ (c) $\bar{\boldsymbol{x}} + 3\sqrt{\lambda_1}\boldsymbol{p}_1$

Figure 3.11.: First mode of variation of a statistical model of the semicircular canals of the ear which has been constructed using the nonrigid mesh registration algorithm presented in this thesis. Spherical or genus 1 parameterization is not applicable to this data set because the shapes are of genus 3.

$f : \mathcal{S} \to \mathbb{R}^3$ is sought such that $f(\mathcal{S}) = \hat{\mathcal{S}}$, where $f(\mathcal{S})$ means f applied to all points on $\mathcal{S}$. The function f must not be an arbitrary, but a smooth function, such that the mapping is topologically consistent.

This function f is determined during registration by minimizing the energy functional

$$E(f;\mathcal{S},\hat{\mathcal{S}}) = E_{\text{sim}}(f(\mathcal{S}),\hat{\mathcal{S}}) + \zeta E_{\text{reg}}(f). \tag{3.33}$$

The first term measures the similarity between $\mathcal{S}$ and $\hat{\mathcal{S}}$. Let $\text{dist}^2(\boldsymbol{v},\mathcal{S},)$ denote the squared Euclidean distance of $\boldsymbol{v} \in \mathbb{R}^3$ to $\mathcal{S}$. As one-to-one correspondence is required, the following symmetric similarity measure is used:

$$E_{\text{sim}}(f(\mathcal{S}),\hat{\mathcal{S}}) = \int_{\boldsymbol{v}\in\mathcal{S}} \text{dist}^2(f(\boldsymbol{v}),\hat{\mathcal{S}})\,\mathrm{d}\mathcal{S} + \int_{\hat{\boldsymbol{v}}\in\hat{\mathcal{S}}} \text{dist}^2(\hat{\boldsymbol{v}}, f(\mathcal{S}))\,\mathrm{d}\hat{\mathcal{S}}. \tag{3.34}$$

The second term in Equation 3.33 is a regularization term which enforces f to be smooth. The free parameter ζ controls the relative importance of the regularization term.

In practice, $\mathcal{S}$ and $\hat{\mathcal{S}}$ are represented by triangle meshes $M = (\mathcal{V},\mathcal{F})$ and $\hat{M} = (\hat{\mathcal{V}},\hat{\mathcal{F}})$ having m and $\hat{m}$ points, respectively. It is also necessary to discretize the function f: It is represented by the displacement field $\boldsymbol{d} = (\boldsymbol{d}_1,\ldots,\boldsymbol{d}_m)$, $\boldsymbol{d}_k \in \mathbb{R}^3$. The notation $M + \boldsymbol{d}$ is used for the mesh with point set $\{\boldsymbol{v}_1 + \boldsymbol{d}_1,\ldots,\boldsymbol{v}_m + \boldsymbol{d}_m\}$ and triangulation $\mathcal{F}$.

Similarity measure. The continuous similarity term $E_{\text{sim}}(f(\mathcal{S}),\hat{\mathcal{S}})$ is approximated by

$$E_{\text{sim}}(M + \boldsymbol{d},\hat{M}) = \sum_{i=1}^{m}\sum_{j=1}^{\hat{m}} a_{ij}\|\boldsymbol{v}_i + \boldsymbol{d}_i - \hat{\boldsymbol{v}}_j\|^2, \tag{3.35}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times\hat{m}}$ is the sparse correspondence matrix that has already been introduced in Section 3.3.2.1 when defining the consistency term used for consistent genus 1 parameterization. However, the cor-

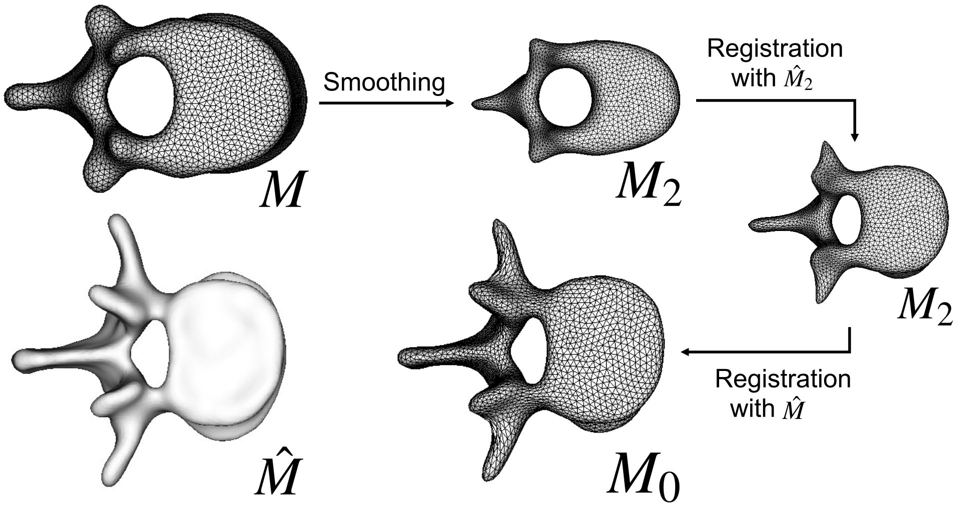


Figure 3.12.: Hierarchic registration of a thoracic vertebra M (Type: Th10) with a lumbar vertebra $\hat{M}$ (L3) using two levels of details. In the first level, a smoothed version M_2 of M is registered with a smoothed version $\hat{M}_2$ of $\hat{M}$. The result M_1 of this registration is registered with $\hat{M}$. The final registration result is M_0.

respondence matrix $\boldsymbol{A}$ is not kept fixed during nonrigid registration, but is iteratively re-estimated, as explained below in Section 3.5.2.

Regularization term. In order to enforce a smooth displacement field $\boldsymbol{d}$, the differences of displacement vectors of adjacent points are penalized. The regularization term is defined by

$$E_{\text{reg}}(\boldsymbol{d}) = \sum_{i=1}^{m} \sum_{j \in \mathcal{N}(i)} \omega_{ij} \|\boldsymbol{d}_i - \boldsymbol{d}_j\|^2. \tag{3.36}$$

The weights ω_{ij} are set in dependence of the distance between $\boldsymbol{v}_i$ and $\hat{\boldsymbol{v}}_j$, that is

$$\omega_{ij} = \frac{\|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_j\|^{-1}}{\sum_{k \in \mathcal{N}(i)} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_k\|^{-1}}. \tag{3.37}$$

Thus, the regularization force is stronger the closer the mesh nodes lie together. The weights are normalized such that $\sum_{j \in \mathcal{N}(i)} \omega_{ij} = 1$ for each $i \in \{1, \dots, m\}$. This means that the sum over all ω_{ij} is m, like in the similarity term.

3.5.2. Multi-scale optimization

The energy functional $E(f)$ is minimized iteratively by alternating between estimating the correspondence relation, that is, re-computing $\boldsymbol{A}$, and minimizing the functional in Equation 3.33 for a given $\boldsymbol{A}$.

This alternating optimization strategy is typical for mesh registration algorithms, and is also used for example in the ICP [BM92] or the Coherent Point Drift [MS10]. Note that when $\boldsymbol{A}$ is fixed, the functional is optimized by minimizing a quadratic function in $\boldsymbol{d}$. This high-dimensional least squares problem is solved using the limited-memory Broyden-Fletcher-Goldfarb-Shannon algorithm [LN89]. The iterative process of re-estimating $\boldsymbol{A}$ and solving a lest-squares problem is repeated until the difference between two consecutive iterations is smaller than a predefined convergence threshold.

Even rigid registration algorithms such as the ICP are not guaranteed to find the optimal registration between two meshes, but may converge to a local optimum. This issue is even more critical for nonrigid registration which has far more degrees of freedom. In particular, poor initial alignment, large difference in size and shape as well as high frequency details affect the robustness of the registration.

In order to account for differences in scale, the meshes are rescaled such that the diagonal of their bounding box is one. The bounding box is aligned with the principal axes of the mesh, not with the coordinate axes, such that the normalization is invariant to the current orientation of the mesh. After scale normalization, a symmetric variant of the ICP algorithm is used to prealign the rescaled meshes. Robustness to high frequency details is achieved by a multi-scale approach: Laplacian mesh smoothing (see, e.g., Bade et al. [BHP06]) is used to compute simplified versions of the original meshes. This step is analogous to low-pass filtering when constructing image pyramids for multi-scale image registration. The nonrigid registration is first performed on the simplified meshes, and finally on the original meshes.

The following listing summarizes the registration of M with $\hat{M}$ for l levels of detail:

1. Compute the meshes $\hat{M}_2,\dots,\hat{M}_l$, where $\hat{M}_1 = \hat{M}$ and $\hat{M}_i$ is obtained from $\hat{M}_{i-1}$ by h iterations of Laplacian smoothing for $i = 2,\dots,l$.
2. Compute M_l by smoothing M with $(l-1)\cdot h$ iterations of Laplacian smoothing.
3. Rescale $\hat{M}_1,\dots,\hat{M}_l$ and M_l such that the diagonal of their principal axes aligned bounding box equals one.
4. Align M_l and $\hat{M}_l$ with a symmetric ICP.
5. For $i = l,\dots 2$
    a) Non-rigidly register M_i with $\hat{M}_i$.
    b) Compute M_{i-1} by adding the deformation field determined in the previous step.
6. M_0 is the final registration result.

An example of the hierarchical nonrigid registration of a thoracic vertebrae with a lumbar vertebrae using two resolution levels is shown in Figure 3.12. The illustration shows that the nonrigid registration is capable of computing large, smooth deformations.

3.5.3. Landmark propagation

As described in Section 2.3.2, the process of establishing correspondence between two shapes $\hat{M} = (\hat{\mathcal{V}},\hat{\mathcal{F}})$ and $M = (\mathcal{V},\mathcal{F})$ can be decomposed into three steps: Landmark definition on $\hat{M}$, registration of $\hat{M}$ with M, and landmark propagation from $\hat{M}$ to M.

1. Landmarks are defined by resampling $\hat{M}$ uniformly using the remeshing algorithm of Fuhrmann et al. [FAKG10]. The result is a (sparse) landmark mesh $M(\hat{\boldsymbol{x}})$.
2. $M(\hat{\boldsymbol{x}})$ is registered with M using the nonrigid alignment algorithm described above.
3. After registration, landmarks are propagated as follows: For each landmark $\hat{\boldsymbol{x}}^i$, the point

$$\hat{\boldsymbol{y}}^i = \frac{1}{\sum_{j=1}^{m} a_{ij}} \sum_{j=1}^{m} a_{ij} \boldsymbol{v}_j \tag{3.38}$$

   is computed, where $\boldsymbol{v}_j \in \mathcal{V}$ for all j. The final landmark $\boldsymbol{x}^i$ is defined as the closest point to $\hat{\boldsymbol{y}}^i$ on M_j. In this case, the closest point search is not restricted to mesh points: The point can be an arbitrary point on the surface.

   The closest point relation may not be consistent with the triangulation of $M(\boldsymbol{x})$. The resulting triangles may have undesired properties, such as sharp angles, and some of triangles may even be inverted. A Laplacian Smoothing-like operation is used to improve the triangulation. For each landmark $\boldsymbol{x}^i$, the point

$$\boldsymbol{y}^i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \boldsymbol{x}^j \tag{3.39}$$

   is computed. The new position of landmark $\boldsymbol{x}^i$ is then defined as the closest point on the surface of M to $\boldsymbol{y}^i$. Note that in contrast to Laplacian Smoothing, this operation ensures that all landmarks lie on M's surface such that details are preserved. Usually, very few smoothing iterations through all landmarks are sufficient to remove all inverted triangles.

3.6. Improving pairwise methods using shape similarity trees

The approach for establishing correspondence using the nonrigid mesh alignment algorithm proposed in the previous section is pairwise, because it always considers two shapes at a time. Similarly, all proposed consistent parameterization methods are inherently pairwise, as consistency is achieved through registration of a mesh to a reference mesh using the ICP.

As discussed in Section 2.3.2.4, the performance of pairwise approaches can be improved by using group information in order to determine a registration order. To this end, it is built upon the shape pre-organizing framework of Munsell et al. [MTSW12] who arrange shapes in a rooted directed tree. Each node in such a tree corresponds to one of the training shapes, and edges only connect nodes if the corresponding shapes are similar.

The shape similarity tree is computed in two steps:

1. A distance matrix $\boldsymbol{D} \in \mathbb{R}^{s \times s}$ is computed that assigns to any pair of shapes a distance d_{ij}. The distance matrix can be interpreted as a fully connected, undirected and weighted graph G.
2. The graph G is condensed to a directed, rooted tree.

The framework is fairly general and various alternatives for its realization are possible. In the following, it is first described how the distance matrix is computed in this work, and then various alternatives for computing the tree are discussed.

3.6.1. Computing distances

Computing distances between pairs of shapes requires some kind of registration. To this end, both Munsell et al. [MTSW12] and Dalal et al. [DSSW10] map the shapes consistently to a common parameter domain, which is either the unit line or the unit circle for the 2D shapes [MTSW12], or the unit sphere in case of 3D [DSSW10]. After the registration, roughly corresponding landmarks are obtained by uniform sampling in the parameter domain. In Munsell's 2D framework, distances are obtained by computing a compromise between the Jaccard Index and the bending energy between the shapes. For 3D brain shapes, Dalal et al. compare the curvature of corresponding points. The use of parameterization is especially in 3D computationally demanding. It also requires different parameterization algorithm in order to handle different topologies.

Instead of choosing Dalal's approach, the symmetric ICP is used in this work for computing distances between shapes without being dependent on their topology: The shapes are aligned, and the symmetric average surface distance between the to point sets is approximated. In order to handle differences in scale and translation, all training shapes are centered and then rescaled such that the diagonals of their bounding boxes have length one. Because the rescaling must be independent of a shape's orientation, its bounding box is aligned with its three principal axes instead of the axes of the world coordinate system. After scale and translation normalization, each pair of shapes is registered with the symmetric variant of the ICP algorithm that has also been used for generating consistent parameterizations in Section 3.2.2 and Section 3.3.2.1.

After alignment, the symmetric average surface distance d_{ij} between the meshes M and $\hat{M}$ is approximated as follows:

$$d_{ij} = \frac{1}{m} \sum_{\boldsymbol{v} \in \mathcal{V}} \|\boldsymbol{v} - \mathrm{C}(\boldsymbol{v}; \hat{\mathcal{V}})\| + \frac{1}{\hat{m}} \sum_{\boldsymbol{v} \in \hat{\mathcal{V}}} \|\hat{\boldsymbol{v}} - \mathrm{C}(\hat{\boldsymbol{v}}; \mathcal{V})\| \tag{3.40}$$

Computing the distance matrix is very time consuming in practice. Its time complexity is in $O(s^2 \cdot R)$, where R is a bound for computing the registration and depends on the size of the shapes and the chosen registration algorithm. Some of the approaches mentioned below for computing the tree have a time complexity in $O(s^3)$, but s is usually considerably smaller than the size of the shapes. This is why tree computation can be done in less than a second for the training sets used within this thesis, while computing the distance matrix often takes several hours.

3.6.2. Computing the similarity tree

Given the distance matrix computed in the previous section, the graph G induced by this matrix is condensed to a rooted, directed tree T. In this thesis, four alternative methods are used to obtain

T from G. Two of these methods, named *MST* and *Dalal*, have been previously used in context of Munsell's framework [MTSW12].

MST: Following Munsell et al. [MTSW12], a minimium spanning tree of G is computed. The tree root is selected to be the node that minimizes the maximal distance to all other nodes in the tree. This node is also called the graph's Jordan center.

Dalal: The heuristic algorithm proposed by Dalal et al. [DSSW10] is used to condense the graph to a tree. Firstly, edges of G are pruned in descending order of their weights until further pruning steps make the graph disconnected. From the pruned graph, the tree is constructed as follows: The tree root is selected to be the node with the most incident edges. The remaining nodes are added iteratively: In every iteration, all nodes adjacent to a tree node are connected to the tree using the connecting edge with minimum weight. The algorithm terminates when all nodes are added to the tree.

JC: The shortest path tree rooted in the Jordan center of G is used as T. This effectively involves solving an all pair's shortest path problem on G in order to find the Jordan center. For this, the Floyd-Warshall algorithm is used [CLRS01].

JC-eps: The method is similar to *JC*, but works on G_ε, a condensed version of G. The graph G_ε contains only the edges of G with weight $\leq \varepsilon$, and ε is chosen to be the smallest value such that G_ε is connected. The use of such an ε-graph has been inspired by the work of Hamm et al. [HYVD10] on nonrigid image registration, in which the registration between two images is decomposed into a series of registrations along similar shapes in a database.

While all four algorithms aim at constructing a tree whose edges connect similar shapes, there is a slightly different reasoning behind every approach. *MST* minimizes the overall registration error, under the assumption that the registration error can be quantified by the dissimilarity measure between two shapes. A drawback is, however, that *MST* may lead to paths with relatively long edge distances to the root node, which means a higher risk that registration errors accumulate on the paths. *Dalal* addresses this problem by trying to construct trees with paths that have a short edge length. The algorithm is of heuristic nature and has no explicitly stated objective that it minimizes. *JC* and *JC-eps* minimize the maximal accumulated registration error, assuming that this error can be quantified using the path length, as measured by the sum of weights of the edges. The use of an ε-graph in *JC-eps* additionally avoids potential registration between dissimilar shapes.

An experimental comparison of the four algorithms on different training sets and using different pairwise approaches is provided in the following Chapter 4.

4. Evaluation of methods for establishing correspondence

This chapter gives a detailed experimental comparison of the new methods for establishing correspondence described in the previous chapter. The main goal of the evaluation is to validate the proposed methods and to find their advantages and disadvantages. An important aspect that is investigated in this chapter is to which degree correspondence influences the segmentation accuracy. To this end, a new evaluation method is used that compares models by how good they perform in model-based segmentation of binary images. The evaluation also contains a detailed investigation of shape similarity trees for improving pairwise correspondence, including the first quantitative comparison of such techniques with groupwise optimization.

The chapter is structured as follows: Section 4.1 introduces a new evaluation methodology which is later used in some of the experiments. Section 4.2 explains the experimental setup: It summarizes which aspects are investigated in the evaluation, describes the data sets that have been used for testing and lists the evaluation methods and measures that have been employed for assessing the model quality. Section 4.3 presents and discusses the results of the comparison of different shape similarity trees, and Section 4.4 presents and discusses the results of the evaluation of the correspondence algorithms.

4.1. Correspondence evaluation by simulating image segmentation

In this section, a new method for evaluating the quality of SSMs is presented. The technique uses image segmentation in order to estimate the generalization ability and specificity of SSMs.

In Section 2.3.4, it has been discussed how generalization ability and specificity is usually evaluated: Shapes are sampled from the learned model distribution and compared with the original training images. The generalization ability can additionally be assessed by leave-one-out tests. In any case, the comparisons require that the shapes are registered, which is inherently achieved through the model construction process that establishes correspondence. While this allows to compare different models quantitatively, it does not provide direct insights to which degree the model quality affects the model's performance in a particular application such as image segmentation.

A second, more subtle problem is that leave-one-out tests for computing the generalization ability are slightly biased when evaluating models constructed with groupwise optimization. Because the optimization is groupwise, the left-out shape influences the correspondence and therefore the SSM, even though it is not used for model learning. It is of course possible to separate the shapes into a training and a test set, such that correspondence is established on the training shapes, and the test shapes are used to measure the generalization ability and specificity. In that case, however, there is no landmark vector available for a test shape, and the model must be registered with the test shape.

The evaluation method proposed here does this model-to-shape registration with the ASM. This approach is especially designed to assess the quality of shape models constructed for image segmentation, because the ASM is frequently used in this application. In order to remove the influence of the image appearance from the tests, not real images, but expert segmentations are segmented, which are represented by binary images.

For adapting the model to the binary image, the following simple appearance model is used: The transition from organ (intensity 1) to background (intensity 0) is modeled by a linear profile $\boldsymbol{f}_{\mathrm{ref}}$ of length $2l+1$ oriented in surface normal direction, which is defined by

$$f_{\mathrm{ref},i} = \begin{cases} 1 & 1 \leq i \leq l \\ \frac{1}{2} & i = l+1 \\ 0 & l+2 \leq i \leq 2l+1. \end{cases} \tag{4.1}$$

In the experiments below, $l = 2$ is chosen. During segmentation, the fitness of a feature $\boldsymbol{f}$ is assessed by computing the distance to $\boldsymbol{f}_{\mathrm{ref}}$, that is $\|\boldsymbol{f} - \boldsymbol{f}_{\mathrm{ref}}\|^2$. Note that $\boldsymbol{f}$ may contain values between 0 and 1 due to interpolation between voxels. After the ASM has terminated, the model quality is assessed with the measures for evaluating the segmentation accuracy that have been discussed in Section 2.6.

Although segmenting a binary image is trivial for intensity-based algorithms, the segmentation computed by the ASM is in general not perfect, because the ASM restricts segmentations to linear combinations of the training examples. Therefore, this procedure estimates the SSM's generalization ability, because a higher segmentation accuracy means that the model adapts better to unseen shapes. In contrast to real image segmentation, the results are not influenced by spuriously selected image features, because binary images have no ambiguous texture information. The evaluation therefore quantifies the segmentation accuracy that can be achieved with a particular SSM given an almost perfect appearance model.

In real image segmentation, a specific SSM preserves the characteristics of an organ's shape even if the image information is misleading or ambiguous. In order to assess the specificity of SSMs, the expert segmentations are perturbed with noise. Noisy binary images are obtained by randomly inverting voxels in the image. The flip probability of a voxel is denoted by p_{flip}, and it is assumed that it is independent from the flip probability of neighboring voxels. By varying p_{flip}, the model's specificity can be assessed under different levels of noise. An experiment with $p_{\mathrm{flip}} = 0$ corresponds to a generalization ability experiment. Moreover, p_{flip} should be smaller than 0.5, because this choice leads to completely random binary images, where each voxel intensity is independently at random drawn from the uniform distribution over $\{0, 1\}$.

It should be noted that segmentation errors can not be exclusively accounted to the limited generalization ability or specificity of the SSM, but also to deficiencies of the model-to-shape registration algorithm, that is, the ASM. In particular, the results may be influenced by the initial placement of the

SSM, as well as by the search strategy of the ASM, which may fail to detect certain image features. The first influence is diminished by repeating a test for a particular test image several times with different initializations. Restrictions of the search strategy can not be as easily removed, but they affect all evaluated models equally. Moreover, they are also present when segmenting real medical images.

4.2. Experimental setup

The following section describes the experimental setup of the tests conducted for evaluating correspondence methods. The main goal of the evaluation is to find out advantages and disadvantages of the various algorithms described in Chapter 3. Probably the most important criterium is the quality of correspondence. However, there are also some other important criteria that must be taken into account, such as how well a method preserves the original input geometry, or how fast a method is. All these aspects are considered in the evaluation. By using the segmentation-based method presented in Section 4.1, the evaluation also gives insights on how important model quality is for segmentation, that is, whether one can expect a significantly better segmentation accuracy just by exchanging the shape model.

Beside the comparison of different correspondence algorithms, the evaluation also provides a comparison of different shape similarity trees. The experiments made by Munsell et al. [MTSW12] and Dalal et al. [DSSW10] have shown that using these trees significantly improves the correspondence quality compared to just choosing a fixed template shape. However, their evaluation is restricted to 2D data sets [MTSW12] or a particular 3D data set [DSSW10], and always uses the same correspondence algorithm. The comparison here uses various 3D data sets, various correspondence algorithms, and also includes two additional methods for computing the tree.

In the following, the data sets, the experiments and the quality measures are explained in more detail.

Data sets. The experiments are conducted on four organ data sets: Livers, prostates, thoracic vertebrae, and lumbar vertebrae. Appendix B provides detailed information about each data set used in the experiments. Only a subset of the available image data for livers and prostates have been used to extract training shapes. For the liver, the subsets LOWRES and SLIVER07-TRAIN have been chosen, which amounts to a total of 47 liver shapes. For the prostate, only the 50 shapes of PROMISE12-TRAIN have been used, because the ground truth of PROMISE12-TEST is not public. Table 4.1 lists additional information about each set of shapes, such as for example the average size of a mesh.

All training shapes have been extracted from expert segmentations of the original image data. The expert segmentations are represented as binary images. An image processing pipeline has been developed in order to convert these binary images into triangle meshes: An image is first isotropically resampled, and a morphological closing operation is used in order to remove small holes in the segmentations. Such holes correspond, for example, to blood vessels in the liver that have not been added to the segmentation by the medical expert. A Gaussian smoothing filter is applied to the image in order to prevent staircase artifacts in the resulting shape. Finally, a mesh is extracted from the processed image

Data set	# Meshes	Topology	Avg. Size (# Points / # Triangles)	Landmark representation (# Landmarks / # Triangles)
Liver	47	Genus 0	9612 / 19220 31239.4[a] / 62474.9[a]	2562 / 5120
Prostate	50	Genus 0	9013.8 / 18023.6	1962 / 3920
Thoracic Vertebra	12	Genus 1	11777.3 / 23554.5	3000 / 6000
Lumbar Vertebra	16	Genus 1	16012.6 / 32025.3	3000 / 6000

[a] Average size of original input meshes before resampling.

Table 4.1.: Data sets used for the evaluation of correspondence algorithms. The table lists the topology of the input shapes, the average size of the original input meshes and the size of the computed landmark vectors.

using Marching Cubes [LC87]. The mesh extraction process described above requires the definition of several parameters, such as the size of the Gaussian kernel used during smoothing, or an appropriate isovalue for Marching Cubes. In order to ensure that the mesh extraction process removes most artifacts, but also maintains the original organ shape, the parameters have been selected interactively for each input image individually: To this end, the quality of each mesh has been inspected visually as well as by considering the average and maximum surface distance between mesh and image representation of the shape.

Additionally, the liver meshes extracted from the image data have been resampled in order to reduce their size. Each mesh is independently mapped to the unit sphere using the method presented in Section 3.2.1. A sparse representation of the shape is then sampled from its parameterization using the icosahedron sampling method (see Figure 3.5 on page 75) with 9612 sampling points.

Experiments. The following correspondence algorithms are evaluated: Alignment of spherical parameterizations (*SphAlign*, Section 3.2.2) and Parameter Space Propagation (*Propagation*, Section 3.2.3); Groupwise optimization of the continous approximation of the MDL function (Equation 2.19) using the aforementioned parameterization methods as initialization (*MDL-SphAlign* and *MDL-Prop*); Consistent genus 1 parameterization (*Genus 1*, Section 3.3.2) as well as subsequent groupwise optimization (*MDL-Genus 1*); and finally, nonrigid registration (*Nonrigid*, Section 3.5). Table 4.2 lists important configuration parameters for each tested correspondence algorithm. Note that identical parameters are used for every data set.

Except for *Nonrigid*, all methods are restricted to shapes of either spherical or toroidal topology, such that they can not be applied to all data sets. The spherical parameterization-based methods *SphAlign*, *Propagation*, *MDL-SphAlign*, *MDL-Prop* are only used for the construction of models of liver and prostate. Likewise, *Genus 1* and *MDL-Genus 1* are only used to construct models for thoracic and lumbar vertebrae. All methods have been configured in such a way that they produce the same number of landmarks for each data set. The number of landmarks used for each data set is listed in Table 4.1.

Because the choice of the template shape influences the quality of models constructed with pairwise registration-based methods, several models are computed for each data set. Each time, another template shape is selected. For lumbar and thoracic vertebrae, each training shape is selected once as template shape. In order to reduce the number of tests, only eleven different template shapes are used for the liver, and only ten different template shapes for the prostate. In these cases, the template shapes are selected uniformly at random without replacement from all shapes in the data set. Additionally, shape similarity trees are constructed for each data set using the algorithms *MST*, *Dalal*, *JC*, and *JC-Eps* described in Section 3.6.2. Each pairwise method is executed on each tree of each compatible data set.

Due to its long computation time, groupwise optimization is only applied to one of the sets of parameterizations generated with the methods *SphAlign*, *Propagation* and *Genus 1* for a particular data set. To this end, the set of parameterizations is chosen that has achieved the best evaluation measures. In every case, this corresponds to a set of parameterization that has been constructed using one of the shape similarity trees.

Measures. A variety of different measures are used to assess the performance of the methods for establishing correspondence. The correspondence quality is measured using the standard criteria specificity (Equation 2.24), generalization ability (Equation 2.26) and compactness (Equation 2.23) that have been introduced in Section 2.3.4. Both generalization ability and specificity are computed using the landmark-based metric for shape comparison. As mentioned in Section 2.3.4, using this metric is problematic if the landmark vectors approximate the original input meshes inaccurately. Therefore, the surface-to-surface distances ASD (Equation 2.50), RMS (Equation 2.51) and HD (Equation 2.52) between a landmark vector and the corresponding input mesh are computed in order to verify that the approximation is sufficiently accurate. These errors are called *representation errors* in the following, or simply ASD, RMS and HD, depending on measure used. For parameterization-based methods, a large representation error is usually caused by distortions in the parameter space. In order to provide a direct measure of these distortions, average and maximum of Degener's energy (Equation 3.13 on page 67) are computed for each generated parameterization.

In addition to the standard approach, generalization ability and specificity are also computed using the segmentation-based method proposed in Section 4.1. The segmentation results are compared with the expert segmentations with the volumetric and surface based measures introduced in Section 2.6.

In order to compare different shape similarity trees for a particular data set, the average and maximum path lengths are computed, where the length of a path is defined by the number of edges it has, and not by the sum of its edge weights. A long path length increases the risk of error propagation.

Comparison to other approaches. The experiments conducted contain only comparisons against one state-of-the-art method, which is groupwise optimization. This choice is justified because groupwise optimization performed favorably in evaluation studies (e.g., [SRN*03]). Because the optimiza-

Table 4.2.: The table shows the configuration parameters used within the experiments for each evaluated correspondence algorithm. Identical parameters are used for all data sets.

Spherical parameterization (*Propagation* and *SphAlign*)				
# CDF reparameterizations[a]	50			
# iterations distortion minimization	200			
Spherical groupwise optimization (*MDL-Prop MDL-SphAlign*)				
Hierarchy levels	4			
Settings per level	Level 1	Level 2	Level 3	Level 4
# Caps	12	42	92	162
Cap radius	0.55	0.27	0.17	0.13
Convergence: max. iterations (per level)	2500			
Convergence: threshold (per level)	0.0001			
Genus 1 parameterization (*Genus-1*)				
Parameter space size $(x \times y)$	1.0×0.55			
Nearest neighborhood size (q)	5			
Max. angle between normals of nearest neighbors	1.5708 rad ($\approx 90^\circ$)			
Regularization parameter (ζ)	0.0001			
Convergence: max. iterations per level	10000			
Convergence: threshold	0.1			
Groupwise optimization on plane (*MDL Genus-1*)				
Parameter space size $(x \times y)$	1.0×0.55			
Hierarchy levels	1			
Cap radius	0.2			
Nonrigid mesh registration (*Nonrigid*)				
Hierarchy levels (l)	3			
# iterations Laplacian smoothing per level	10			
Nearest neighborhood size (q)	5			
Max. angle between normals of nearest neighbors	1.5708 rad ($\approx 90^\circ$)			
Regularization parameter (ζ)	1			
Convergence: maximum # iterations	1000			
Convergence: threshold	0.00001			

[a] In case of propagation, CDF reparameterization is only applied to the template mesh.

tion is initialized using one of the parameterization methods developed in this thesis, the question may arise how this compares to other initialization methods. A study [KGWB11] conducted in cooperation with the *Institute of Medical Engineering*, Universität zu Lübeck, has shown that groupwise optimization performs better when using either alignment of spherical parameterizations or an earlier version of Parameter Space Propagation [KW10] than when using SPHARM [KSG99], which is used by other authors to initialize correspondence optimization. In a different study [KW10], it has been shown that the earlier version of Parameter Space Propagation performs better than the algorithm of Davies et al. [DTT06] for computing consistent parameterizations. Finally, it has been shown that consistent genus 1 parameterization constructs models of higher quality than just using the ICP [BKFW11].

4.3. Comparison of shape similarity trees

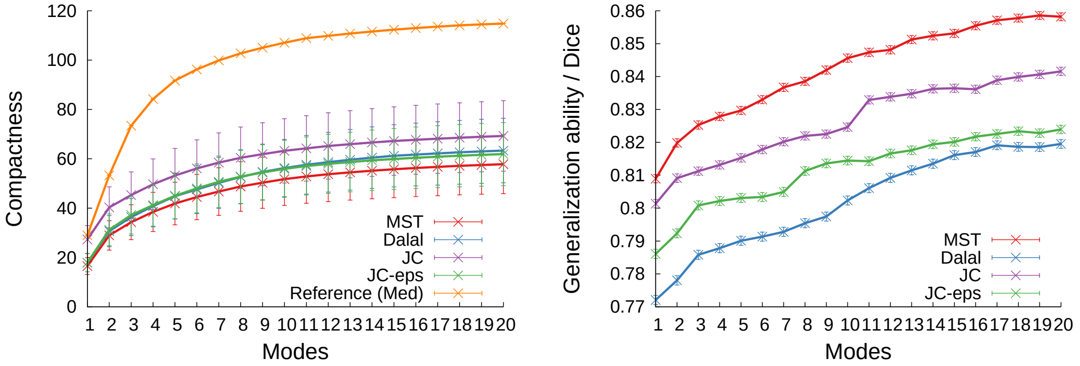
4.3.1. Results and observations

Table 4.3 shows the average and maximum path length of the shape similarity trees computed for each of the four data sets. One can observe that *MST* produces by far the longest path lengths: The maximum path length is often two times, in one case even five times longer than that of other trees. Likewise, *MST* trees have longer average path lengths than trees computed with the other three methods. The path lengths of trees computed with *Dalal*, *JC* and *JC-eps* are comparable. *Dalal* achieves the shortest average path length on the liver and the prostate data set, respectively, but also produces a rather degenerated tree on the lumbar data set: The maximum path length of six is rather high for a tree with only 16 nodes.

A general observation is that using shape similarity trees has a positive impact on the model quality compared to using a fixed template shape. In most experiments, the models constructed using shape similarity trees achieve considerably better evaluation measures. An example for this shows Figure 4.1(a), which plots the compactness of liver shape models constructed with *Nonrigid* on different trees. The only two cases in which choosing a single template shape is better than using a tree is on the prostate, using either *Nonrigid* or *Propagation* as correspondence algorithm. Here, using a template shape leads to more compact models than when using the *MST* shape similarity tree. However, the

Table 4.3.: Average and maximum path length of the shape similarity trees computed with different methods on the four data sets. The path length is measured by the number of edges.

	Liver		Prostate		Lumbar		Thorax	
	Average	Maximum	Average	Maximum	Average	Maximum	Average	Maximum
MST	3.7	8	4.5	10	3.6	6	2.6	5
Dalal	1.3	3	1.1	2	2.9	6	1.6	3
JC	1.9	3	1.5	3	1.5	3	1.4	3
JC-eps	1.8	4	1.4	3	2.5	5	1.8	4



(a) Liver Compactness (Nonrigid)

(b) Liver - Generalization, Segmentation (SphAlign)

Figure 4.1.: Evaluation of the impact of shape similarity trees on the quality of liver models. (a): Compactness of liver models constructed with *Nonrigid* using different similarity trees. The yellow curve shows the median model compactness obtained when using a template shape instead of a tree. Lower values are better. (b): Segmentation-based generalization ability of liver models constructed with *SphAlign* and different similarity trees. Larger values are better.

template shape-based models are still worse than models computed using a similarity tree computed with *Dalal*, *JC* or *JC-eps*.

The experimental results do not yield a unique ranking of the four methods for computing shape similarity trees. Rather, the ranking depends both on the data set and the correspondence algorithm. On the liver data set, *MST* achieves the overall best results when combined with *SphAlign* or *Nonrigid*, while *Dalal* performs worst here. Especially for *SphAlign*, exchanging the liver similarity tree has a particularly large influence on the segmentation accuracy, as illustrated in Figure 4.1(b): The Dice overlap achieved by the model constructed using *MST* is almost 5% larger than that of *Dalal*. On the other hand, the *MST* liver similarity tree performs worst when combined with *Propagation*.

On the prostate, models constructed with *Dalal*, *JC* and *JC-eps* do better than those constructed with *MST* when considering the standard measures. However, this does not lead to a significant difference in performance in the segmentation-based evaluation: All methods perform here almost equally well. When using *Nonrigid* to build shape models for lumbar or thoracic vertebrae, exchanging the shape similarity tree does not change the model quality significantly. In contrast to that, the choice of the tree matters when using *Torus* for model building, especially on lumbar vertebrae. Here, *Dalal* and *JC-eps* perform consistently better than *MST* and *JC*.

JC-eps is arguably the most robust of the methods: In most experiments, it either achieves the best results, or is almost as good as the best ranked method. Despite its similarity to *JC*, *JC-eps* does better in most experiments.

4.3.2. Discussion

The experimental results show clearly that the use of shape similarity trees improves the model quality significantly. In almost all cases, models constructed with shape similarity trees are better than those based on a single template shape. These results confirm previous studies on shape similarity trees on 2D [MTSW12] and 3D shapes [DSSW10]. In contrast to these two studies, not only one, but four different correspondence algorithms are used. This means that it can be excluded with great certainty that the benefit of using similarity trees is restricted to particular correspondence algorithms. Moreover, the results are confirmed on four distinct data sets.

While it is evident that using shape similarity trees is superior to just using a single template shape, the experimental results do not suggest a particular method for computing the tree: Changing either the data set or the correspondence algorithm often changes the ranking of the different tree computation methods.

As already observed by Dalal et al. [DSSW10], *MST* produces trees with rather long path lengths, which introduces a higher risk of error propagation. In fact, *MST* performs significantly worse than the other methods on the prostate. However, it also achieves better results than the other methods on the liver when using *Nonrigid* or *SphAlign* as correspondence algorithm, even though the average and maximum path length of the *MST* tree are twice as long as on the other trees. This indicates that long path lengths are not necessarily a disadvantage.

JC, *JC-eps* and *Dalal* construct trees with similar path lengths. If short path lengths are desirable, for example because error accumulation is a known issue with the correspondence algorithm used, choosing any of the three methods is appropriate. Although the experimental results are not definite enough to allow for a confident recommendation, there are several arguments that speak in favor of using *JC-eps*. Firstly, *JC-eps* arguably achieves the best results when considering all experiments, because it performs in most experiments equally well or only slightly worse than the best method. Secondly, *JC-eps* explicitly enforces short propagation paths because the shape similarity tree is a shortest path tree. In contrast to that, the criteria for constructing the tree used in *Dalal* are heuristic. The degenerated tree that *Dalal* builds on the lumbar vertebrae set indicates that this method does not guarantee to produce short path lengths, even though it has been explicitly designed to do so. Finally, *JC-eps* avoids registration of very dissimilar shapes by pruning edges with large weight from the tree. Such registrations can not be completely excluded when using *JC*.

4.4. Comparison of correspondence algorithms

4.4.1. Results and observations

In the following, the evaluation results of the correspondence algorithms are listed for each data set individually.

Table 4.4.: Running time of correspondence algorithms on the liver data set.

Algorithm	Hardware[a]	Running time				
		total	w/o prepr.[b]	for template	per registration	
Shape similarity tree	C1	3.8 h				
SphAlign	C1	4.9 h	70 m	90 s	89 s	SD: 2.1
Propagation	C1	4.6 h	51 m	91 s	65 s	SD: 0.7
MDL-SphAlign	C2	630 h	625 h			
MDL-Propagation	C2	618 h	613 h			
Nonrigid	C1	4 h	17 m	22 s	21 s	SD: 6.9

[a] see Appendix B.4

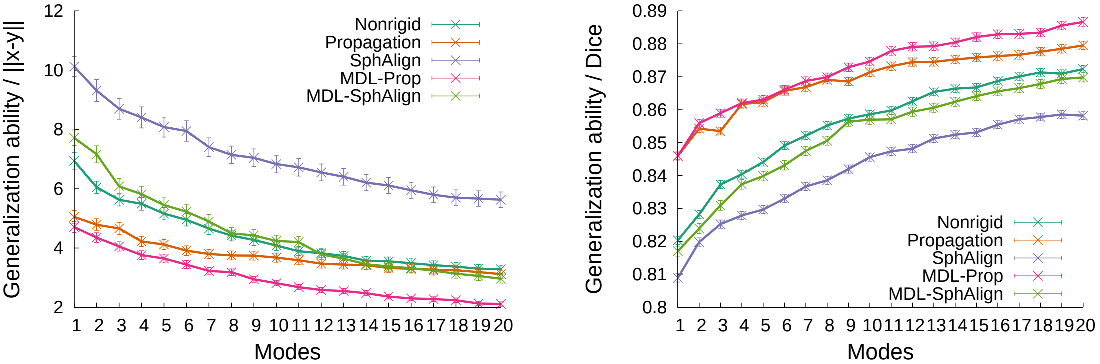
[b] without preprocessing (shape similarity tree computation; parameterization for groupwise optimization.)

4.4.1.1. Liver

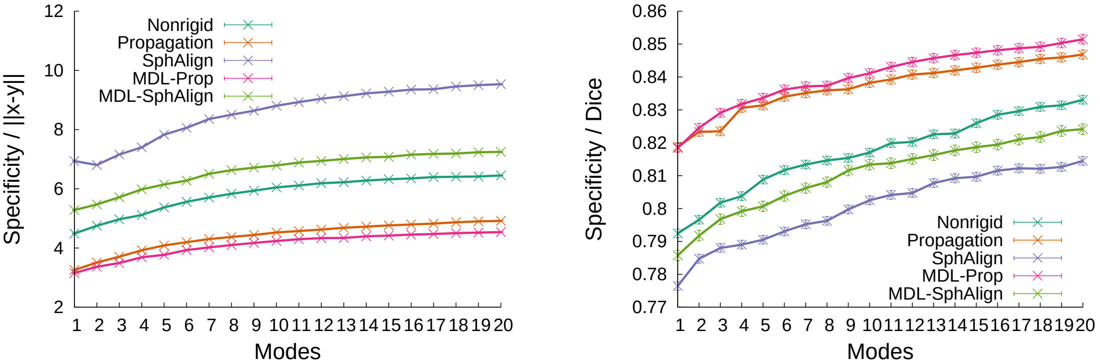
Figure 4.2 shows the performance measures of the evaluated correspondence algorithms applied to the liver data set. The results of the five correspondence quality measures shown in Figures 4.2(a)-(e) are consistent: *MDL-Propagation* achieves the best values in all measures. Its initialization, *Propagation*, performs only slightly worse. The liver model constructed with *Nonrigid*, which is ranked third, is considerably less compact than those built with the two aforementioned methods. It also achieves a significantly smaller Dice coefficient in the segmentation-based measures, especially if only few modes of variations are used. But also for 20 modes of variation, the Dice coefficient is about 1.5 to 2 % worse than that of *MDL-Propagation*. The two methods based on rigid alignment of spherical parameterizations, *SphAlign* and *MDL-SphAlign*, achieve the overall worst results.

One can observe that the relative difference between initialization and optimized model, as measured by the correspondence quality metrics, is much higher for *SphAlign* and *MDL-SphAlign* than for *Propagation* and *MDL-Propagation*: For example, the compactness of the *MDL-SphAlign* model with 20 modes is about one third smaller than that of the corresponding *SphAlign* model, whereas the corresponding *Propagation* and *MDL-Propagation* models have roughly the same compactness (Figure 4.2(e)). The development of the MDL objective function during optimization for both initializations is shown in Figure 4.3(a): The objective value decreases by approximately 60 % for *MDL-Propagation*, and about 230 % for *MDL-SphAlign*. *MDL-Propagation* has always a lower objective value when comparing corresponding iterations, but the final objective value of *MDL-SphAlign* is lower than the initial objective value of *MDL-Propagation*.

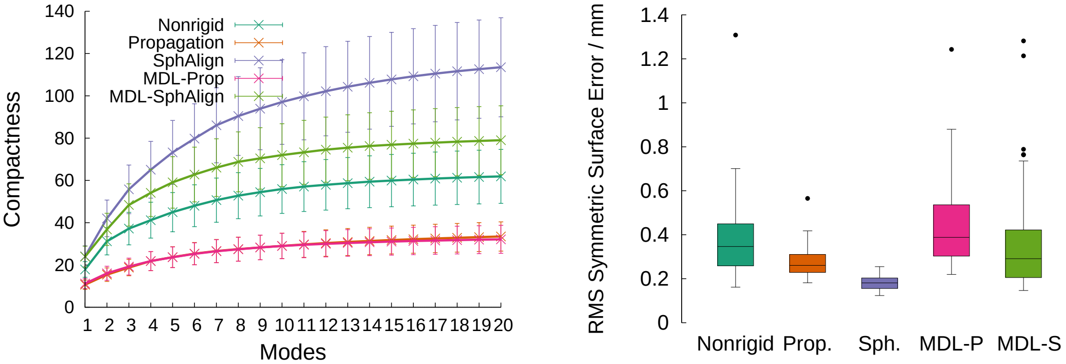
As shown in Figure 4.2(f), the RMS representation error is always smaller than 1 mm. The representation error differs significantly for different methods: *SphAlign* has the smallest error, and *MDL-Propagation* the largest. The HD is always below 4 mm for *SphAlign* (Median HD: 2.1 mm). The worst HD of a sample is 23.8 mm and has been constructed with *MDL-SphAlign* (Median HD: 4.6 mm). In



(a) Liver - Generalization ability (b) Liver - Generalization ability (Segmentation)

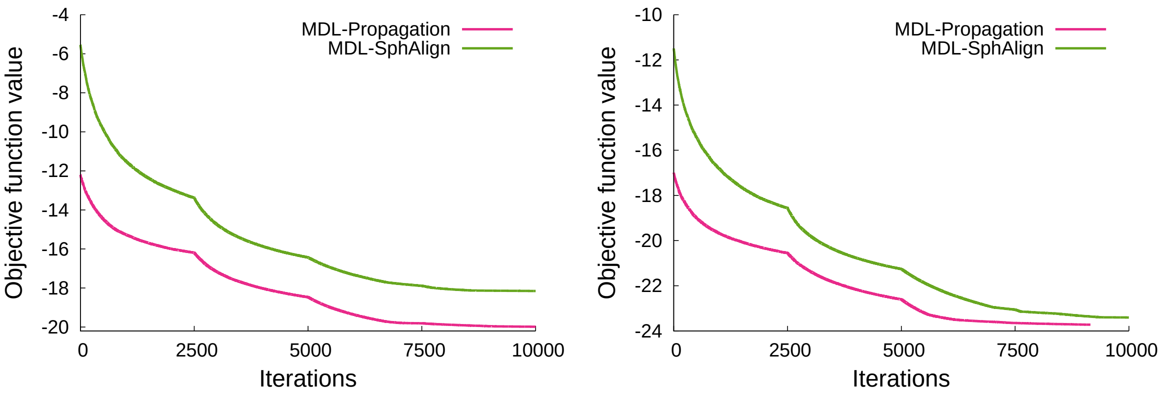


(c) Liver - Specificity (d) Liver - Specificity (Segmentation)



(e) Liver - Compactness (f) Representation error

Figure 4.2.: Performance measures of the methods for establishing correspondence on the liver data set. Note that higher values indicate better quality for the segmentation-based measures used in Figures (b) and (d), while for all other measures, smaller values are better.



(a) Liver (b) Prostate

Figure 4.3.: Development of the objective value of the MDL function during correspondence optimization on the liver (a) and prostate (b) data set, using different initializations. After 2500, 5000 and 7500 iterations, the algorithm switches to a finer resolution level with more localized reparameterizations. On the prostate, *MDL-Propagation* satisfied the convergence criterion after 9154 iterations.

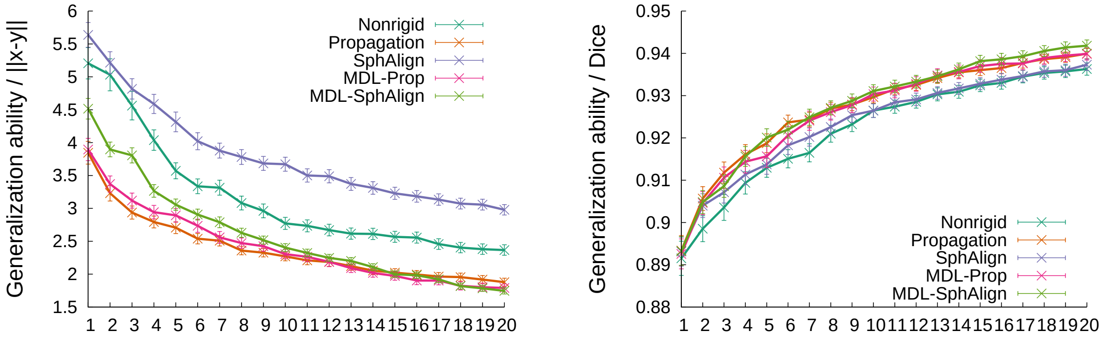
general, it can be observed that groupwise optimization increases the error compared to the input parameterizations computed with *Propagation* and *SphAlign*.

Figure 4.4 shows the running time of the algorithms on the liver data set. All pairwise registration-based methods require less than five hours to establish correspondence. Most of the running time is spent for computing the shape similarity tree. The best overall running time has *Nonrigid*, which requires approximately four hours in total. In contrast to that, correspondence optimization requires more than 25 days, even though the computation has been done with faster hardware.

4.4.1.2. Prostate

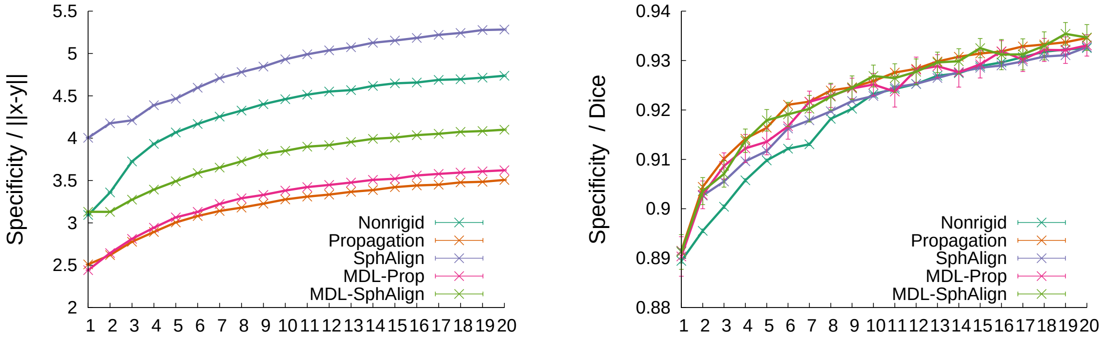
Figure 4.4 shows evaluation results of the correspondence algorithms on the prostate data set. *MDL-Propagation* and *Propagation* generate the most compact models, and additionally have the smallest errors in the standard specificity (Figure 4.4(c)) and generalization ability tests (Figure 4.4(a)). The difference between both methods is very small. *MDL-SphAlign* achieves a similarly good generalization ability for models with more than ten modes of variation, but is less compact and has a higher error in the standard specificity tests. *Nonrigid* and *SphAlign* perform significantly worse than the other three methods with respect to the standard measures.

When comparing the groupwise optimization approaches *MDL-Propagation* and *MDL-SphAlign*, similar observations can be made as for the liver data set: The gain in model quality achieved by optimization is much higher for *MDL-SphAlign* than for *MDL-Propagation*. Nevertheless, the objective value changes in both cases considerably during optimization, as shown in Figure 4.3(b). Again, the



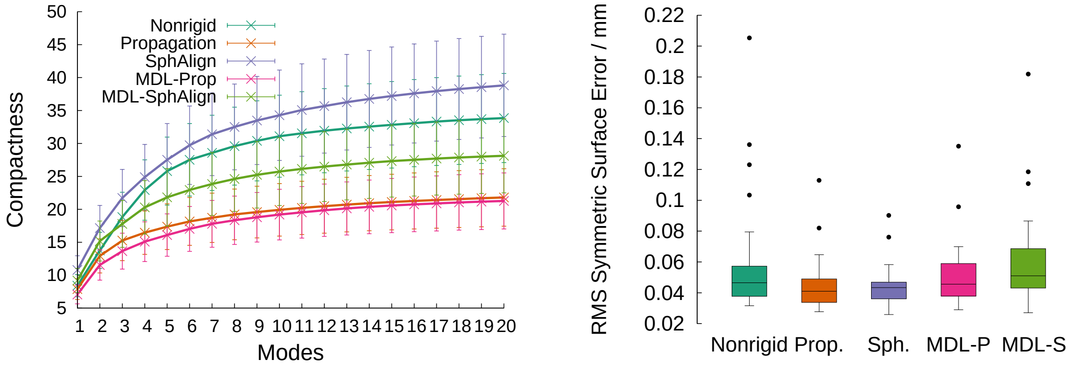
(a) Prostate - Generalization ability

(b) Prostate - Generalization ability (Segmentation)



(c) Prostate- Specificity

(d) Prostate - Specificity (Segmentation)



(e) Prostate - Compactness

(f) Representation error

Figure 4.4.: Performance measures of the methods for establishing correspondence on the prostate data set. Note that higher values indicate better quality for the segmentation-based measures used in Figures (b) and (d), while for all other measures, smaller values are better.

Table 4.5.: Running time of correspondence algorithms on the prostate data set.

Algorithm	Hardware[a]	Running time				
		total	w/o prepr.[b]	for template	per registration	
Shape similarity tree	C1	4.6 h				
SphAlign	C1	5.7 h	65 m	51 s	52 s	SD: 49.2
Propagation	C1	5.5 h	52 m	77 s	62.29 s	SD: 8.8
MDL-SphAlign	C2	557 h	551 h			
MDL-Propagation	C2	473 h	468 h			
Nonrigid	C1	4.7 h	86 s	9 s	1.6 s	SD: 0.8

[a] see Appendix B.4

[b] without preprocessing (shape similarity tree computation; parameterization for groupwise optimization.)

objective value of *MDL-SphAlign* is always higher than that of *MDL-Propagation* in corresponding iterations, but the final objective of *MDL-SphAlign* is lower than the initial value of *MDL-Propagation*.

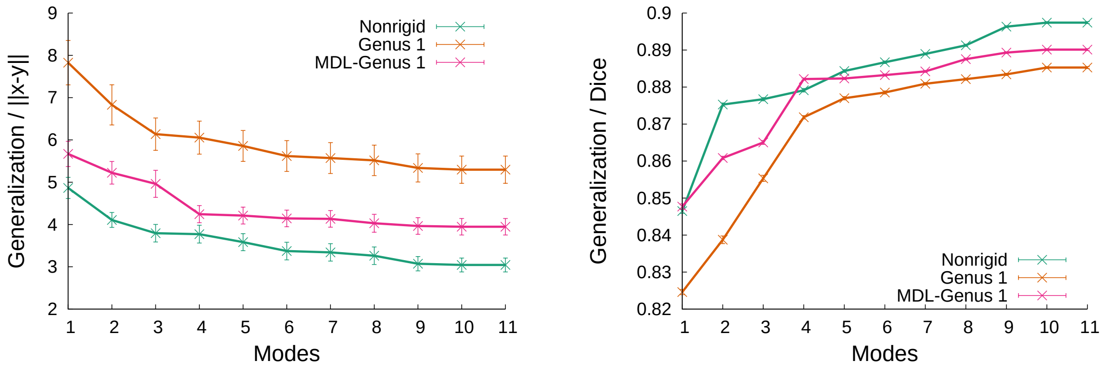
All five methods achieve similar values in the segmentation-based evaluation. *Nonrigid* and *SphAlign* perform slightly worse results than the other three methods, but the difference is smaller than 1%.

For all methods, the RMS representation error is smaller than 0.02 mm. The best results were achieved using *SphAlign* and *Propagation*, and the difference between these methods is negligible. With groupwise optimization (*MDL-SphAlign* and *MDL-Propagation*), the error increases slightly. *Nonrigid* has slightly higher errors than the consistent parameterization methods and slightly smaller errors than the groupwise optimization methods in average. However, it also has the most outliers with respect to the representation error, including the sample with the highest HD in total (4.1 mm). The median HD is below 0.5 mm for all methods.

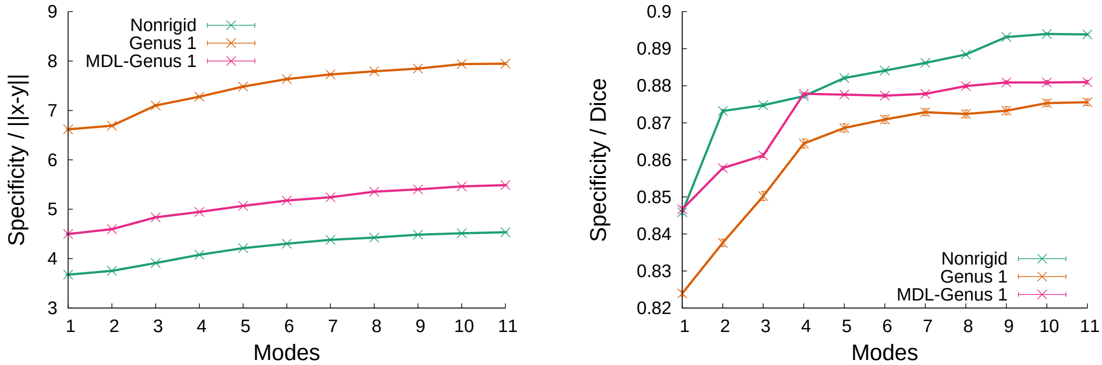
The running time is less than six hours for all three pairwise registration-based methods. The running time is dominated by shape similarity tree computation, which takes approximately four hours and a half. Given the tree, *Nonrigid* is the fastest pairwise method, and needs only 86 seconds to construct landmark vectors of all 50 input shapes, while *SphAlign* and *Propagation* require both approximately an hour. Correspondence optimization is by far the slowest method and takes between 19 and 23 days.

4.4.1.3. Thoracic vertebrae

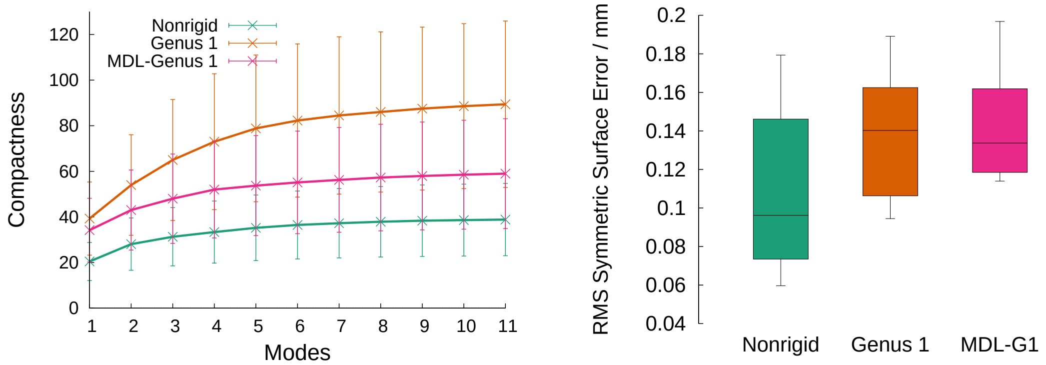
The correspondence measures yield a consistent ranking of correspondence algorithms applied to thoracic vertebrae, as illustrated in Figures 4.5(a)-(e). *Nonrigid* achieves the best results in all measures, and *Genus 1* the worst. Groupwise optimization improves the quality of the *Genus 1* model significantly: The compactness of the optimized model is only half as big as that of the initial parameterization (Figure 4.5(e)). Likewise the generalization ability and specificity measured with the conventional approach decrease by approximately 50 % (Figures 4.5(c)-(d)). Using segmentation-based measures,



(a) Thoracic vertebrae - Generalization ability (b) Thoracic vertebrae - Generalization ability (Segm.)



(c) Thoracic vertebrae- Specificity (d) Thoracic vertebrae - Specificity (Segmentation)



(e) Thoracic vertebrae - Compactness (f) Representation error

Figure 4.5.: Performance measures of the methods for establishing correspondence on thoracic vertebrae. Note that higher values indicate better quality for the segmentation-based measures used in Figures (b) and (d), while for all other measures, smaller values are better.

Table 4.6.: Running time of correspondence algorithms on the thoracic vertebrae data set.

Algorithm	Hardware[a]	Running time				
		total	w/o prepr.[b]	for template	per registration	
Shape similarity tree	C1	17 m				
Nonrigid	C1	21 m	4 m	14 s	20 s	SD: 7.7
Genus 1	C1	138 m	120 m	1220 s	546 s	SD: 256.5
MDL-Genus 1	C2	24 h	22 h			

[a] see Appendix B.4

[b] without preprocessing (shape similarity tree computation; parameterization for groupwise optimization.)

Nonrigid achieves an overlap which is by approximately 1.5–2 % higher than that of *Genus 1* when using the 11 modes of variation. *MDL-Genus 1* lies in between.

The RMS representation error of the samples is shown in Figure 4.5(f). It is below 0.2 mm for all methods. *Nonrigid* has in average the smallest RMS, but also tends to have higher HD than the other two methods. The highest overall HD have *Nonrigid* and *MDL-Genus 1* with about 3 mm. In contrast to the liver and the prostate data set, the sample quality does not degrade as much when using groupwise optimization. *MDL-Genus 1* even reaches a lower median RMS than *Genus 1*, but has in most cases a higher HD.

The running times of the algorithms are listed in Table 4.6. *Nonrigid* is the fastest of all methods and requires 21 minutes, which includes 17 minutes for computing the shape similarity tree. *Genus 1* takes about an hour, and groupwise optimization (*MDL-Genus 1*) about a day.

4.4.1.4. Lumbar vertebrae

On the lumbar vertebrae data set, similar observations can be made as for thoracic vertebrae. As shown in Figures 4.6(a)-(e), *Nonrigid* achieves the best results for all correspondence measures, and *Genus 1* the worst. Like it is the case for thoracic vertebrae, the standard correspondence measures decrease by about 50 % when optimizing the initial parameterization computed with *Genus 1* using groupwise optimization. The overlap measured by the segmentation-based evaluation is about 1 % higher for *Nonrigid* than for *Genus 1* when using 15 modes of variation. However, the optimized model is almost as good as *Nonrigid* (Figures 4.6(b) and (d)).

As illustrated in Figure 4.6(f), the RMS representation error of the samples is below 0.3 mm for all correspondence algorithms. *Nonrigid* has the highest RMS and HD in average. It also produces the sample with the worst HD with about 6.5 mm. The median HD is below 4.5 mm for *Nonrigid*, and below 2.5 for *Genus 1* and *MDL-Genus 1*. Groupwise optimization does not significantly increase the representation error compared to the initial parameterization.

Table 4.7.: Running time of correspondence algorithms on the lumbar vertebrae data set.

Algorithm	Hardware[a]	Running time				
		total	w/o prepr.[b]	for template	per registration	
Shape similarity tree	C1	15 m				
Nonrigid	C1	23 m	8 m	15 s	32 s	SD: 8.9
Genus 1	C1	115 m	100 m	248 s	382 s	SD: 247
MDL-Genus 1	C2	38 h	37 h			

[a] see Appendix B.4

[b] without preprocessing (shape similarity tree computation; parameterization for groupwise optimization.)

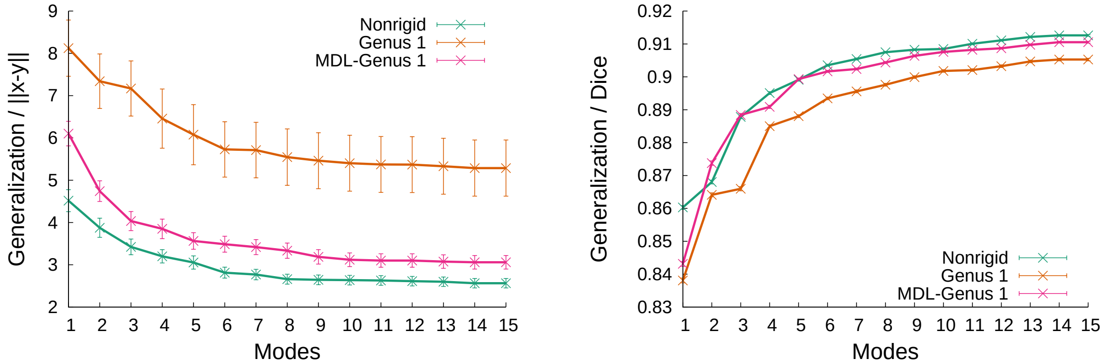
The running times of the algorithms are listed in Table 4.7. *Nonrigid* is the fastest method and requires about 23 minutes to establish correspondence on the 16 input shapes. Most of the running time, in total 15 minutes, is spent for computing the shape similarity tree. *Genus 1* takes about 2 hours, and *MDL-Genus 1* one day and a half.

4.4.2. Discussion

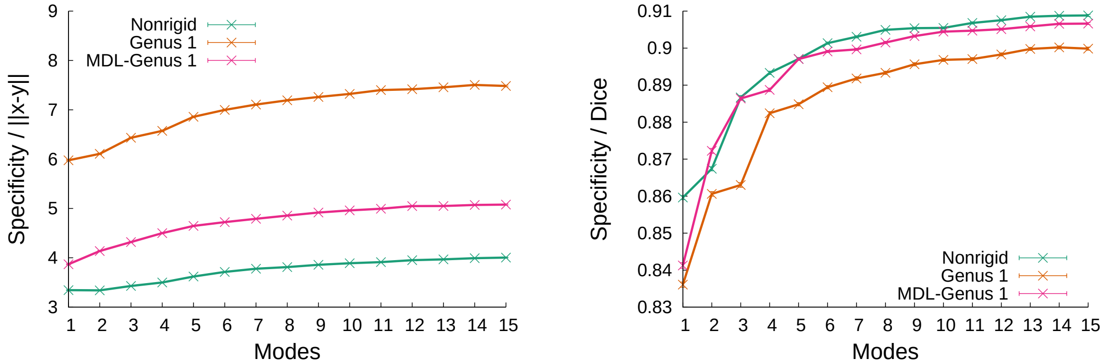
Among the pairwise approaches for shapes with spherical topology, *Propagation* constructs the best models. The constructed liver and prostate models are far more compact than that built using *SphAlign* or *Nonrigid*. On the liver, *Propagation* also achieves a significantly higher overlap in the segmentation-based evaluation. The main drawback of this algorithm is that several parts of it are heuristic: The fuzzy point correspondence relation and the correction strategy that accounts for triangle flips on the parameterization. The fuzzy point correspondence relation can be easily replaced by a probabilistic model, for example that of the EM-ICP [GP06]. The heuristic correction strategy is more problematic, because it is possible that it can not remove all overlapping or flipped triangles on shape classes with very high intra-class variability. However, given the good model quality *Propagation* achieves on the liver, which is arguably an organ with very high shape variability, one can expect that it will also perform well on most other organ shapes of spherical topology.

The second consistent spherical parameterization approach, *SphAlign*, performs significantly worse than *Propagation*. Compared to *Propagation*, *SphAlign* has a stronger focus on distortion minimization than on consistency. Because consistency is accomplished by a global rigid alignment of the parameter spaces, no additional area distortions are introduced. On the downside, this also means that the correspondences are less accurate than that of *Propagation*. *SphAlign* might be a reasonable choice for data sets with very large shape variability, on which *Propagation* might to fail because of its heuristic nature.

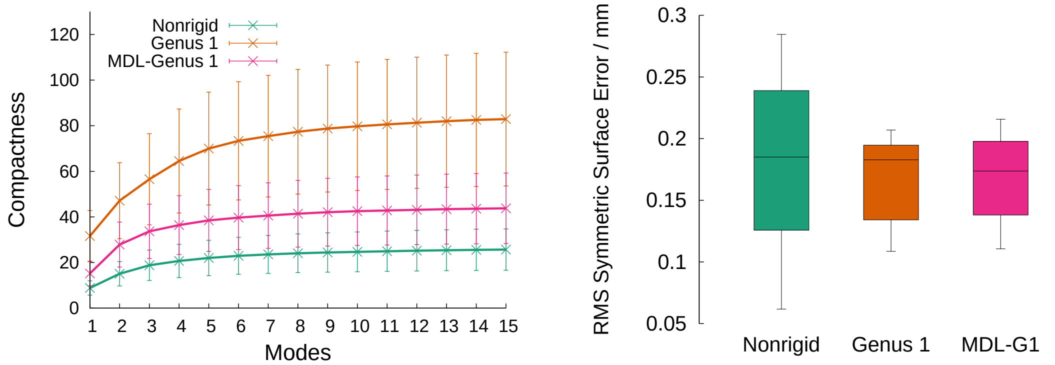
In the experiments, the quality of *Propagation* and *SphAlign* models could be improved with subsequent groupwise optimization on both the liver and the prostate data set. Although the objective



(a) Lumbar vertebrae - Generalization ability (b) Lumbar vertebrae - Generalization ability (Segm.)



(c) Lumbar vertebrae- Specificity (d) Lumbar vertebrae - Specificity (Segmentation)



(e) Lumbar vertebrae - Compactness (f) Representation error

Figure 4.6.: Performance measures of the methods for establishing correspondence on lumbar vertebrae. Note that higher values indicate better quality for the segmentation-based measures used in Figures (b) and (d), while for all other measures, smaller values are better.

Table 4.8.: Comparison of different methods for establishing correspondence. The symbols mean poor (--), bad (-), medium (○), good (+) and very good (++).

Method	Topology (genus)	Running time	Quality of Correspondence
Propagation	0	+	++
SphAlign	0	+	○
Groupwise optimization (*MDL-**)	0, 1	--	+
Genus 1	1	○	○
Nonrigid	arbitrary	++	+

value changed considerably during optimization, *Propagation* and *MDL-Propagation* models differ only slightly in the evaluation measures. In case of *SphAlign*, a slightly better segmentation accuracy can be achieved. Considering the long optimization time, it is questionable whether these small improvements justify the long optimization times. In fact, the evaluation results show that pairwise registration based-approaches such as *Propagation*, when combined with shape similarity trees, can construct models with comparable quality as explicitly optimized models.

It is also interesting to note that the final objective value of *MDL-SphAlign* is lower than that of *Propagation* on both liver and prostate data set, while the *Propagation* model achieves better performance measures. This shows that a smaller description length does not necessarily imply a better model quality. Finally, one can argue that all correspondence quality measures are somewhat biased towards *MDL*, because both *MDL* and the measures are based on the assumption that a linear shape model is used.

Genus 1 allows for consistent parameterization of shapes of toroidal topology. An advantage of this approach compared to *Nonrigid* is that it guarantees topological consistency. On the downside, it performs worse in the correspondence quality metrics. A reason for this might be that the current parameterization approach is not flexible enough, because distortion and consistency are enforced by local, not global optimization. Thus, there is a risk that the optimization runs into a poor local optimum. It is therefore to be expected that the approach can be improved by employing a global optimization scheme, for example by using B-Splines or CPS reparameterization, which would affect more than one point at a time.

Subsequent groupwise optimization using *MDL-Genus 1* has a positive impact on the correspondence quality. On lumbar vertebrae, the segmentation accuracy becomes almost as good as that of *Nonrigid*. It should be noted that no multi-resolution optimization strategy has been employed for the planar domain. From the development of the objective function on the spherical domain shown in Figure 4.3, one can see that the objective function decreases more rapidly immediately after switching to a finer resolution level. This suggest that a multiresolution optimization strategy will also improve the results of groupwise optimization on the plane.

Among the evaluated methods, *Nonrigid* is the most versatile approach. It does not rely on parameterization, is therefore not restricted to a particular topology and is applicable to any of the data set used for evaluation in this thesis. It is also applicable to thin structures such as the pancreas, for which spherical parameterization does not achieve satisfactory results because of unavoidable, high distortions. A pancreas model constructed with *Nonrigid* is illustrated in Figure 2.2 on page 15. In the evaluation, *Nonrigid* achieves the best results on thoracic and lumbar vertebrae, but it also performs well on prostate and liver. Finally, *Nonrigid* is the fastest algorithm among the proposed methods.

Table 4.8 gives a brief summary of the evaluation of different methods and compares them with respect to their versatility, running time and quality of correspondence.

5. The Probabilistic Active Shape Model

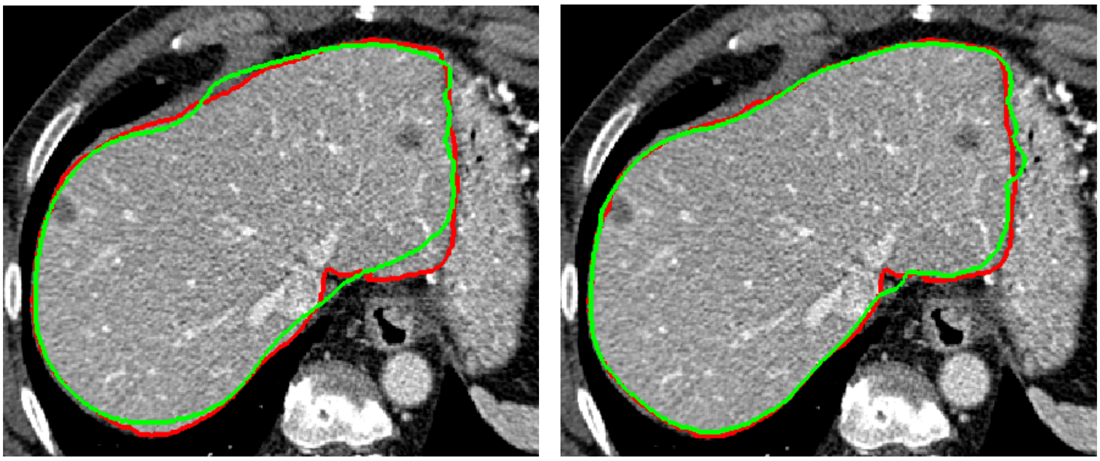
Segmentation of medical images is often hampered by noise, partial volume effects and low contrast between neighboring structures. For this reason, model-based algorithms such as the ASM [CTCG95] and its variants are frequently used for segmentation (see Section 2.4). The strength of the ASM is that it prevents leaking of the segmentation into neighboring structures by imposing constraints on the shape of the segmentation. The algorithm is generic and versatile and can be applied to virtually all organs that have a characteristic shape. As discussed in Section 2.8, the ASM has two major weaknesses: **Limited flexibility** and **limited specificity**. This chapter presents an improved ASM, the *Probabilistic Active Shape Model*, which addresses both limitations of the original approach.

In contrast to the original ASM, the Probabilistic ASM does not impose hard, but rather soft constraints on the shapes. Deformations beyond the patterns observed in the training data are allowed as long as they are sufficiently supported by image information. This increases the flexibility of the Probabilistic ASM and thereby allows for a better adaptation of the model to previously unseen shape instances, as illustrated in Figure 5.1. This additional flexibility is crucial for accurate image segmentation because the organ in an image usually has details that are not present in the training data, either because the number of training examples is too small or not representative enough, or because the organ has atypical, pathological variations.

The Probabilistic ASM allows seamless integration of either linear or nonlinear shape priors. Previously proposed linear and nonlinear ASM variants do not allow to treat different types of shape models in a unified way. Even worse, nonlinear ASMs that have been described in the literature so far (see Section 2.2.4) have some serious practical limitations: For example they do not account properly for the loss of information that occurs when projecting high dimensional data to a low-dimensional feature space, which is especially important when working with very high dimensional 3D shapes. The Probabilistic ASM overcomes these shortcomings, and provides a unified approach for both linear and nonlinear shape priors.

The Probabilistic ASM follows the same iterative, EM-like approach of the original ASM by alternating between image feature search and imposing shape constraints. The difference to the standard approach is how these constraints are imposed: Instead of a simple least-squares projection of a deformed shape to the subspace spanned by the first principal components, shape constraints are enforced by minimizing a probabilistic energy which determines a compromise between image information and prior knowledge.

In Section 5.1, the generic energy used within the Probabilistic ASM is introduced. The energy consists of three different terms, which are described in the following sections in more details: Section 5.2 treats the global shape energy, which is defined by either a linear or a nonlinear shape model. Sec-



(a) Standard ASM (b) Probabilistic ASM

Figure 5.1.: Liver segmentation using the standard ASM (a) and the linear Probablistic ASM (b): Because the ASM restricts shapes to the shape model's linear subspace, it is not flexible enough to delineate details of previously unseen liver shapes. The Probabilistic ASM is not restricted to this subspace, and thus achieves a higher accuracy.

tion 5.3 introduces a local shape term which ensures smoothness of the segmentation boundary. The final term, the image energy, which draws the shape model to detected image features, is discussed in Section 5.4.

5.1. Energy formulation

The Probabilistic ASM uses energy minimization for imposing shape constraints in each iteration. Given a deformed shape $\hat{\boldsymbol{x}}$, which has been obtained by displacing landmarks to detected image features, the constrained shape is determined by minimizing the energy

$$E(\boldsymbol{x};\hat{\boldsymbol{x}},\boldsymbol{w}) = \alpha \cdot \left[E_{\text{image}}(\boldsymbol{x};\hat{\boldsymbol{x}},\boldsymbol{w}) + E_{\text{local}}(\boldsymbol{x})\right] + E_{\text{shape}}(\boldsymbol{x}). \tag{5.1}$$

The energy consists of three forces: Image energy, local shape energy and global shape energy. A free parameter α is used to balance the global shape energy with the other two energies.

The image energy $E_{\text{image}}(\boldsymbol{x};\hat{\boldsymbol{x}},\boldsymbol{w})$ draws the model to the detected image features, which are represented by $\hat{\boldsymbol{x}}$. The vector $\boldsymbol{w} \in \mathbb{R}^n$ contains a weight $w_i \geq 0$ for the feature of each landmark i. Intuitively, a higher weight encodes a higher certainty that the image feature really lies on the sought organ boundary. The actual definition of the weights is task of the used appearance model, which differs depending on the application. As explained in detail in Section 5.4, the weights w_i are normalized before energy minimization in order to achieve an automatic balancing of image energy and local shape energy. The local shape energy $E_{\text{local}}(\boldsymbol{x})$ is a regularization force that enforces smooth organ boundaries. It ensures that a landmark can not move far away from its neighbors. Finally, the global shape energy constrains shapes to be similar to the training shapes. This similarity is measured by the negative log-likelihood of a probability distribution learned from the training shapes. It is possible to use either a linear shape energy, in which case the probability distribution is a multivariate Gaussian, or a nonlinear shape energy, in which case the probability distribution is a multivariate Gaussian in the feature space defined by the selected kernel function.

Similar to other ASM variants (see Section 2.4.4), the shape constraints are imposed in the coordinate system of the SSM, not in the image coordinate system. This means that the so-called pose parameters—rotation, translation and scaling—are treated as external parameters. They are estimated prior to energy minimization by superimposing the model's mean mesh with the deformed shape $\hat{\boldsymbol{x}}$ using the method of Horn [Hor87]. It is important that during energy minimization, the pose parameters must not be changed in order to guarantee that the constrained is correctly mapped back to the image. As explained in Section 5.3, the local shape energy is a shrinking force, such that explicit measures have to be taken to maintain the scale of the shape.

Instead of treating pose parameters as external parameter, it would also be possible to explicitly optimize pose during energy minimization, as it is done in the Diffusion Snake framework of Cremers et al. [CTWS02]. As discussed in Section 2.5.2, treating pose as external parameters allows a much faster computation of the gradients, especially in case of high dimensional 3D shapes.

5.2. Global shape energy

In ASMs, leaking of the segmentation into neighboring structures is prevented by imposing bounds on the shape of the segmentation with a SSM, such that the shape of the segmentation is similar to the training shapes. The way shape constraints are enforced is crucial for the generalization ability and specificity of the segmentation algorithm. Too over-restrictive shape constraints impede a high delineation accuracy, because one can not expect that the SSM captures every natural, let alone pathological, shape variation. If the trained SSM is unspecific, because it considers a far broader class of shapes plausible than those similar to the training shapes, it is not able to ensure that the segmentation is similar to the training shapes, and thus may not effectively prevent leaking into neighboring structures.

The Probabilistic ASM framework enforces shape constraints with the global shape energy. The shape is decomposed into a weighted sum of several deformation patterns—for example the eigenvectors in case of the linear shape model—and an energy is assigned to each pattern. Patterns that have been frequently observed in the training data have a low energy compared to rare patterns. However, evens patterns that have never been observed in the training data are in principle allowed—albeit they generate higher energies than any observed pattern. For computing the energy, the concept of Moghaddam and Pentland [MP97] is used, which has already been discussed in Section 2.2.2: The energy is split into the distance in feature space (DIFS) and the distance from feature space (DFFS). By integrating the DFFS into the energy, shapes are allowed to deviate slightly from the patterns extracted during learning. This is a crucial difference to the standard ASM, where the deformations are restricted to observed patterns, and additional deformations are impossible.

In the following paragraphs, the linear shape energy based on PCA and a nonlinear shape energy based on Kernel PCA are presented. The notation used here has been introduced in Section 2.2.2 for linear SSMs and in Section 2.2.4 for nonlinear SSMs.

5.2.1. Linear shape energy

In case of a linear SSM learned using PCA, the underlying model assumptions is that the shapes are multivariate Gaussian distributed. The approximation of the Mahalanobis distance (see Equation 2.12) of this high-dimensional Gaussian is used as shape energy:

$$E_{\text{shape}}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{t}\frac{b_i}{\lambda_i} + \frac{1}{2\rho}\|\boldsymbol{r}\|^2. \tag{5.2}$$

As discussed in Section 2.2.2.1, the optimal choice for ρ from the point of view of information theory is the average of the discarded eigenvalues (Equation 2.13). In all practical applications in this thesis, the number of training examples is far smaller than their dimension, that is $s \ll 3n$. Because the number of nonzero eigenvalues can not exceed $s-1$, ρ is defined to be

$$\rho = \sum_{i=t+1}^{s-1}\frac{\lambda_i}{s-t-1}. \tag{5.3}$$

The difference to Equation 2.13 is in the divisor, which is $s-t-1$, not $3n-t$. This avoids that ρ becomes extremely small compared to λ_t, which would restrict the model's flexibility.

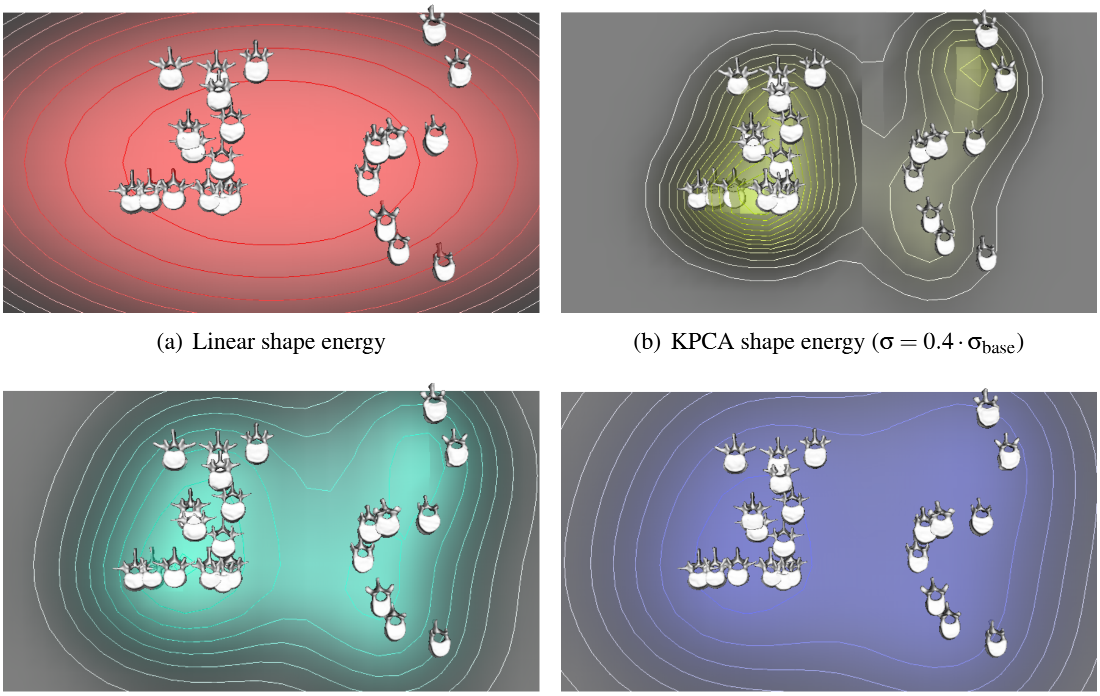
5.2.2. Nonlinear shape models using Kernel PCA

Nonlinear global shape energies are realized using Kernel PCA, which has been discussed in Section 2.2.4.3. In contrast to the linear shape energy, Kernel PCA allows to model multimodal shape energies, which can be beneficial for example for shape classes which can be partitioned into several subclasses. Figure 5.2 gives a visual comparison of the linear shape energy of a data set of vertebrae with several KPCA-based nonlinear shape energies learned from the same data.

The nonlinear energy is conceptually very similar to the linear shape energy. However, instead of using the approximated Mahalanobis distances in $\mathbb{R}^{3n}$, the approximated Mahalanobis Distance in the kernel feature space is used:

$$E_{\text{shape}-\text{kpca}}(\boldsymbol{x}) = \sum_{i=1}^{t}\frac{\beta_i^2}{\nu_i} + \frac{1}{\varepsilon}\left(\tilde{k}(\boldsymbol{x},\boldsymbol{x}) - \sum_{i=1}^{t}\beta_i^2\right). \tag{5.4}$$

As already defined in Section 2.2.4.3, the β_i are the kernel principal components, and the ν_i are the kernel eigenvalues. Similar to the corresponding term in the linear shape energy, the second term



(a) Linear shape energy

(b) KPCA shape energy ($\sigma = 0.4 \cdot \sigma_{\text{base}}$)

(c) KPCA shape energy ($\sigma = 0.7 \cdot \sigma_{\text{base}}$)

(d) KPCA shape energy ($\sigma = \sigma_{\text{base}}$)

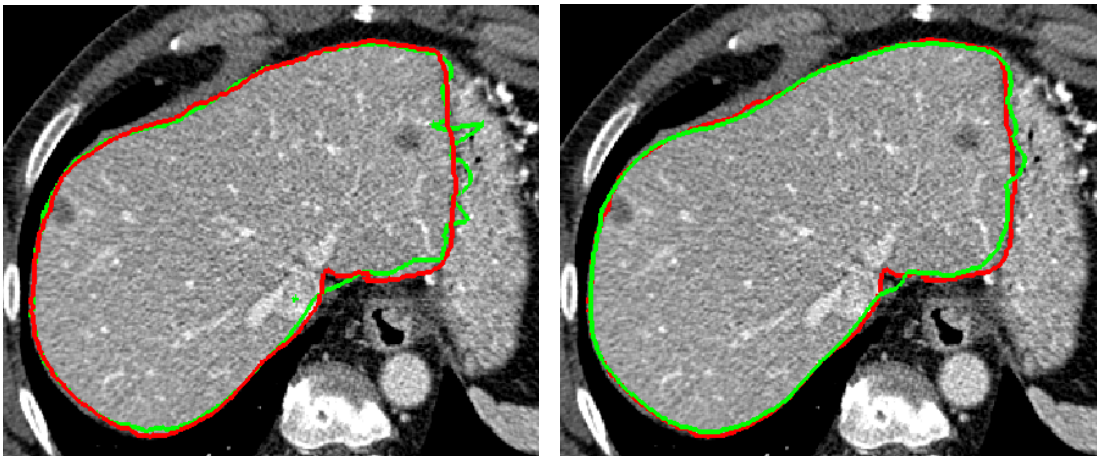
Figure 5.2.: Comparison of different shape energies for a data set of vertebrae. For visualization purposes, the energies with multidimensional domain have been sampled and visualized on the first two principal components spanned by the training data. The training examples have also been projected to their two principal components. Shown are the linear shape energy (a), and KPCA-based nonlinear shape energies (b)-(d) with varying scaling parameter σ.

penalizes the loss of information when projecting the shape to the space spanned by the principal axes in kernel feature space. This nonlinear shape has also been used by Cremers et al. in their 2D Diffusion Snakes framework [CKS03].

In order to determine the dimension of the kernel principal subspace, first a regularization parameter ε is defined, and then all eigenvectors are discarded with $\nu_i \leq \varepsilon$. In all practical experiments, $\varepsilon = 0.001$ is chosen. The remaining r eigenvalues are renormalized such that $\frac{1}{\sqrt{\nu_i}} \gamma_i^T \gamma_i = 1$. This has the effect that if a linear kernel $k_{\text{lin}}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$ is chosen, and $\varepsilon = \rho$, Equation 5.4 and Equation 5.2 are identical up to a constant factor of 2.

Any kernel function can be used in Kernel PCA. In this work, only Gaussian radial basis function kernels of the form

$$k_{Gauss}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\sigma^2}\right) \tag{5.5}$$



(a) Without local energy (b) With local energy

Figure 5.3.: Liver segmentation using the Probabilistic ASM without (a) and with local shape energy (b). Without local shape energy, the segmentation contour can become jagged in regions with ambiguous image information.

are used. This kernel function has a free hyperparameter σ which controls its scale. As illustrated in Figure 5.2, the energy function becomes smoother the larger σ is. Because any positive real value can be used for σ, innumerable different distributions can be realized. It is not obvious which choice of σ is optimal for a particular data set. In order to simplify parameter selection, σ is specified in this work in dependence of properties of the given training set. Inspired by Cremers et al. [CKS03],

$$\sigma_{\text{base}} = \sqrt{\frac{1}{S}\sum_{i=1}^{S} \min_{j\in\{1,\ldots,S\},j\neq i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2} \tag{5.6}$$

is used as a base for parameter selection. That is, instead of choosing σ directly, a scaling factor σ_{scale} for σ_{base} is chosen, such that $\sigma = \sigma_{\text{scale}} \cdot \sigma_{\text{base}}$.

5.3. Local shape energy

In the previous section, global linear and nonlinear shape energies have been described. Both allow deformation patterns that can not be expressed through the model's (kernel) eigenvectors. It is ensured that these patterns cause a higher energy than any learned deformation patterns by penalizing the projection error, or DFFS. It is important to note that this technique allows to control the magnitude, but not the nature of additional variation. A consistent movement of neighboring landmarks towards the true organ boundary may cause the same DFFS as a single landmark that is drawn towards an outlier that has been spuriously selected by the appearance model. The latter kind of deformation generates jagged, and therefore implausible segmentation contours. Therefore, additional means are necessary to regularize this additional flexibility. This regularization is done using the local shape energy, which explicitly enforces smooth organ contours.

The problem of jagged segmentation contours is illustrated in Figure 5.3: The segmentation in Figure 5.3(a) has been computed without local shape energy, that is, by combining the global linear shape energy with the image energy $E_{\text{image-lsqr}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w})$ defined in Equation 5.9 in the next section. While the model (green contour) adapts well to the liver boundary manually delineated by experts (red contour) in regions where the image contrast is high, it becomes extremely jagged in ambiguous regions with low contrast between liver and surroundings. It is possible to avoid such jagged regions by reducing the influence of the image energy by decreasing α, but this comes also at the cost of a lower overall segmentation accuracy. The alternative strategy, which is taken in this work, is to introduce an additional energy that enforces smoothness: As visible in Figure 5.3(b), the use of a local regularization force maintains the accuracy in regions with high contrast, and effectively smooths the contour in ambiguous regions.

The local shape energy explicitly introduces the assumption into the model that segmentation contours must be smooth. The basic idea is to penalize edge lengths of the model mesh. This approach has two major problems: Firstly, it is non-adaptive, which means that it treats every region in the model mesh equally. This is a problem because shapes typically have some characteristic high curvature regions that must not be smoothed away. Secondly, meshes with a smaller global scale also have a smaller edge length. Thus, penalizing edge lengths also lets the mesh shrink, which is undesired because the energy must not affect the shape's scaling.

The first problem can be tackled by integrating prior knowledge about the mesh's curvature into the local energy term. This is done by considering the mean relative position of landmark i to its neighbors in the training data, which is

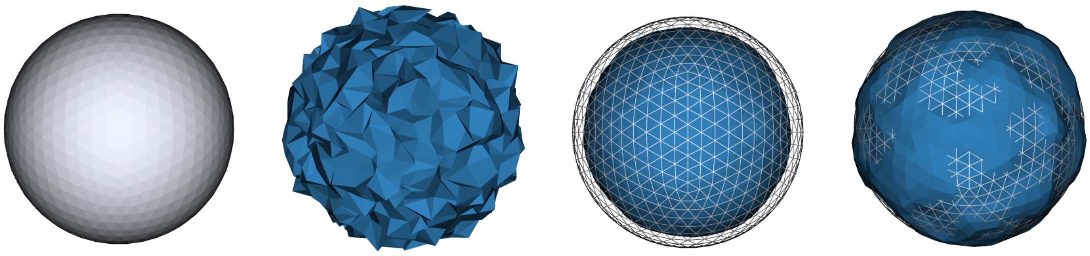
$$\boldsymbol{\mu}^i = \bar{\boldsymbol{x}}^i - \frac{1}{\mathcal{N}(i)} \sum_{j \in \mathcal{N}(i)} \bar{\boldsymbol{x}}^j. \tag{5.7}$$

Instead of penalizing the deviation of a landmark to the mean of its neighbors, which is equivalent to penalizing edge lengths, the local energy penalizes deviation of landmark from $\boldsymbol{\mu}^i$. With this, the local energy can be defined by

$$E_{\text{local}}(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} \|\boldsymbol{x}^i - \boldsymbol{\mu}^i - \boldsymbol{x}^j\|^2. \tag{5.8}$$

The vectors $\boldsymbol{\mu}^i$ build the *local shape model* of the Probabilistic ASM; the use of this local model avoids that characteristic high curvature features are smoothed away. It is important to note that this simple local model can be directly extracted from the model's mean mesh and does not require the use of any other additional parameters. It is possible to extend the model to incorporate higher order statistics, for example, by penalizing deviations from $\boldsymbol{\mu}^i$ adaptively in dependence of the variance of this deviations observed in the training data.

The local model energy still has the problem that it tends to prefer meshes with short edges. This problem is handled by the image energy.



(a) Sphere (b) Sphere with noise (c) $E_{\text{image-lsqr}} + E_{\text{local}}$ (d) $E_{\text{image}} + E_{\text{local}}$

Figure 5.4.: A sphere (a) is modified by randomly displacing points along their surface normals (b), which simulates deformation caused by the ASM's image feature search. The deformed sphere defines the vector $\hat{\boldsymbol{x}}$. (c): If an optimization is started that combines the local energy with the least squares image energy, $E_{\text{image-lsqr}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w})$, the optimized shape is considerably smaller than the original sphere. (d): If $E_{\text{image}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w})$ is used as image force, the scale is maintained.

5.4. Image energy

The task of the image energy is to ensure that the constrained shape $\boldsymbol{x}$ determined during optimization is close to the detected image features described by the vector $\hat{\boldsymbol{x}}$. An obvious choice for the image energy is to penalize the sum of weighted squared distances between landmarks and corresponding image features, which is

$$E_{\text{image-lsqr}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w}) = \sum_{i=1}^{n} w_i \|\boldsymbol{x}^i - \hat{\boldsymbol{x}}^i\|^2. \tag{5.9}$$

The weights $w_i \geq 0$ are determined by the appearance model. The image energy has the advantage that it is simple, intuitive and, as it is a quadratic function, easy to optimize. However, it can not compensate the shrinking force generated by the local shape energy described in the previous section. This is illustrated in the toy example shown in Figure 5.4, where an optimization is used to smooth a noisy sphere, while enforcing closeness to the noisy data through the image energy.

Therefore, a different shape energy is used in the Probabilistic ASM which does not only penalize deviations from $\boldsymbol{x}$, but also counteracts the shrinking force of the local shape energy. The idea is to penalize the net deviation of $\boldsymbol{x}^i$ from neighboring landmarks: This means that the movement of a landmark is compensated by a movement of neighboring landmarks in the opposite direction, as illustrated in Figure 5.5. Thus, if a landmark is pulled towards the shape's center of gravity through the local energy, the neighboring landmarks are pulled in the other direction and thereby avoid a shrinking of the shape. Formally, the image energy is defined as

$$E_{\text{image}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w}) = \sum_{i=1}^{n} \|w_i(\boldsymbol{x}^i - \hat{\boldsymbol{x}}^i) + \sum_{j\in\mathcal{N}(i)} w_j(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j)\|^2. \tag{5.10}$$

Note that the optimum of both $E_{\text{image}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w})$ and $E_{\text{Image-lsqr}}(\boldsymbol{x};\hat{\boldsymbol{x}};\boldsymbol{w})$ is $\boldsymbol{x} = \hat{\boldsymbol{x}}$.

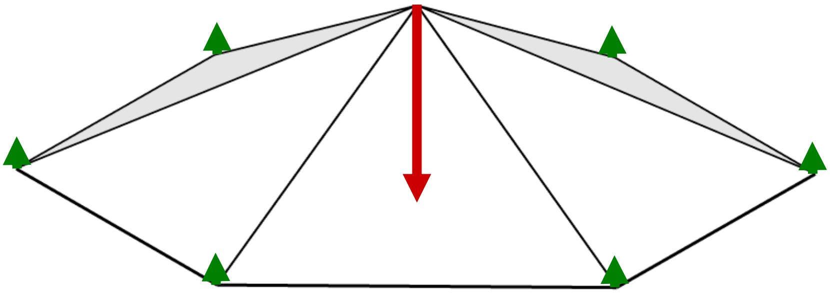


Figure 5.5.: Illustration of the image energy: The displacement of a point from its image feature is compensated by a displacements of the neighbors in the opposite direction.

Normalization of image feature weights The energy of the Probabilistic ASM is comprised of three energies which must be balanced against each other. Although three energies are used, the formulation of the energy given in Equation 5.1 only uses a single balancing parameter, which balances image energy and local shape energy with the global shape energy. The reason that only a single balancing parameter is used is because practical experiments show that image energy and local shape energy can be sufficiently well balanced by normalizing the weights w_i provided by the appearance model. The basic idea of the weight normalization is to ensure that local energy and image energy have the same influence. This is done by normalizing the weights such that $\sum_{i=1}^{n} w_i^2 = n$.

5.5. Optimization

Finding the shape that minimizes the energy in Equation 5.1 is a high dimensional optimization problem which must be solved in each iteration of the Probabilistic ASM. For fast segmentation, an efficient optimization method is required. For the experiments made in this thesis, the limited-memory Broyden-Fletcher-Goldfarb-Shannon algorithm (L-BFGS) [LN89] is used for energy minimization. The L-BFGS belongs to the class of *quasi-Newton* optimization methods which exploit both gradient and curvature information. Unlike Newton's method, however, quasi-Newton methods only approximate the Hessian matrix of second derivatives. In the L-BFGS, the Hessian is approximated by a few vectors, which is particularly time- and memory-efficient for high-dimensional optimization problems.

The approximation of the Hessian in the L-BFGS is based on a history of updates in the preceding iterations and the current gradient. The gradients of the energy function is computed analytically. To ease the re-implementation of the Probabilistic ASM, the gradients of the most important energies are specified in Appendix A. An implementation of L-BFGS is available in the open source library VXL[1], which has also been used for the implementation made for this thesis.

Because linear shape energy, local shape energy and image energy are quadratic functions, the energy has a unique minimum in case of linear shape models. Kernel PCA based shape energies can

---

[1] http://vxl.sourceforge.net

be multimodal, such that their combination with local shape energy and image energy is potentially multimodal as well. In this case, it is only guaranteed that the optimization converges to local, but not necessarily to a global optimum.

5.6. Comparison to related approaches

This section compares the Probabilistic ASM to related segmentation algorithms that use explicit shape representation. In particular, similarities and differences to other ASM variants, such as the standard ASM [CTCG95], Shape Constrained Deformable Models [WKL*01], and ASMs with nonlinear shape priors are elaborated. The approach is also compared with the Diffusion Snakes framework [CTWS02, CKS03], which uses the same shape energies as the Probabilistic ASM.

5.6.1. Active Shape Model

The difference between the Probabilistic ASM and the standard ASM [CTCG95] is in the way shape constraints are imposed. As explained in detail in Section 2.4.4, the ASM maps a high dimensional landmark vector via a matrix multiplication to a low dimensional parameter space vector to enforce shape constraints. In contrast to that, the Probabilistic ASM uses an energy minimization approach which balances between image information and prior knowledge. One fundamental difference between the approaches is that the technique used in the ASM does not readily extend to nonlinear shape models, while the approach in the Probabilistic ASM does. In the following, the discussion is therefore restricted to linear shape models.

The matrix multiplication used in the ASM (Equation 2.28) enforces that every plausible shape must be within the model's subspace: Thus, the DFFS of every shape allowed by the ASM is always zero. Furthermore, hard constraints are used that restrict principal components to a certain interval: This means that the ASM follows the idea of a 'space of plausible shapes', in which a shape is either plausible or not—two plausible shapes are equally likely, and two implausible shapes equally unlikely. In contrast to that, the Probabilistic ASM rather searches for a compromise between image information and prior knowledge about the object's shape: No shape is considered as implausible, as long as it is supported sufficiently by the image energy. Shape deformation outside the model's subspace is possible, but this kind of deformation has a lower probability (i.e., causes a higher energy) than deformation with the same magnitude along any of the model's principal axes. The property that the algorithm allows this extra-variability is the key to achieve more accurate segmentations than the standard ASM. The higher accuracy of the Probabilistic ASM is shown in Chapter 6 in quantitative experiments on real data.

The standard ASM does not use a local model, nor does it explicitly require shapes to be smooth. This is arguably because there is no need for an explicit smoothness assumption if shapes are restricted to a subset of all linear combinations of the training shapes. If the training shapes are smooth, then linear combinations of the training shapes are usually sufficiently smooth as well. However, if the

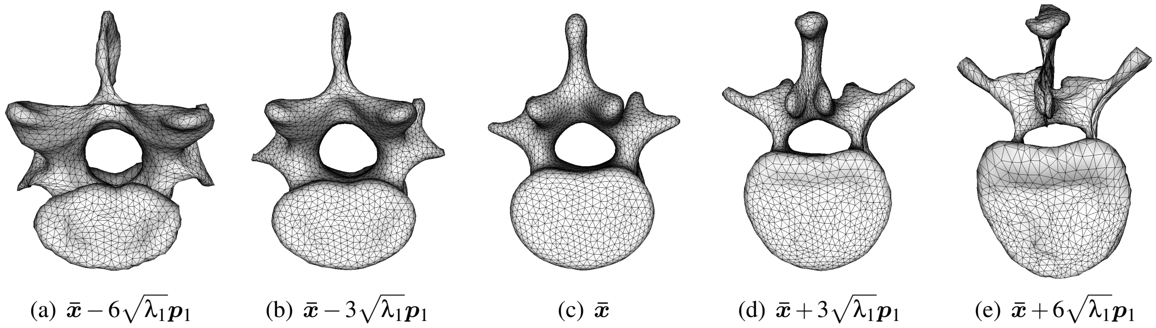

(a) $\bar{\boldsymbol{x}} - 6\sqrt{\lambda_1}\boldsymbol{p}_1$ (b) $\bar{\boldsymbol{x}} - 3\sqrt{\lambda_1}\boldsymbol{p}_1$ (c) $\bar{\boldsymbol{x}}$ (d) $\bar{\boldsymbol{x}} + 3\sqrt{\lambda_1}\boldsymbol{p}_1$ (e) $\bar{\boldsymbol{x}} + 6\sqrt{\lambda_1}\boldsymbol{p}_1$

Figure 5.6.: Illustration of the first mode of variation of a linear lumbar vertebra model. Assuming that the first principal component is restricted to lie within the interval $[-3\sqrt{\lambda_1}, 3\sqrt{\lambda_1}]$, as it is usually done, the standard ASM considers shapes (b)-(d) as plausible, and shapes (a) and (e) as implausible. SCDMs assign zero shape energy to each of the shown shapes. The Probabilistic ASM assigns zero energy to shape (c), an energy of 9 to shapes (b) and (d), and an energy of 36 to shapes (a) and (e).

model's subspace is left, as it is done in the Probabilistic ASM, explicit means are necessary to enforce smoothness.

An important property of the ASM's mapping of landmark vectors $\hat{\boldsymbol{x}}$ to principal components $\boldsymbol{b}$ (Equation 2.28) is that this projection inherently preserves all information on the model's principal axes. In other words, the constrained shape $\boldsymbol{x}$ is the nearest shape to $\hat{\boldsymbol{x}}$ within the model's subspace. Thus, the approach inherently guarantees similarity to $\hat{\boldsymbol{x}}$. On the other hand, the shape energy of the linear Probabilistic ASM has a unique minimum, which is the mean shape. Similarity to $\hat{\boldsymbol{x}}$ must be explicitly enforced using a data term—the image energy. Although the idea of using a data term may occur trivial, some proposed approaches for nonlinear ASM segmentation actually ignore it.

There are free parameters both in the ASM and in the Probabilistic ASM: A common parameter is the dimension t of the principal subspace. Moreover, in the ASM, interval bounds on the principal components must be selected which separate between plausible and implausible shapes. For both parameters, there are rules of thumbs that usually work well in practice. In contrast to the ASM, the Probabilistic ASM requires a parameter for balancing image and shape energy. There is no obvious choice for this parameter, which means that the effort for choosing correct parameters is higher for the Probabilistic ASM than for the standard ASM.

5.6.2. Shape Constrained Deformable Models

Shape Constrained Deformable Models (SCDMs) [WKL*01], which have been introduced in Section 2.4.4.2, are a flexible variant of ASMs with linear shape priors. SCDMs and its variants have been used by many authors for medical image segmentation. Like the Probabilistic ASM, SCDMS impose shape constraints with energy minimization. In both algorithms, the shape energy allows for more flexible segmentation than the standard ASM because shapes can leave the principal subspace spanned by the shape model's eigenvectors.

In SCDMs, the DIFS is not penalized. Every shape within the model's subspace has zero shape energy, even shapes that are arbitrarily far away from the mean. From experience, shapes whose principal components b_i are outside the interval defined by three standard deviations $[-\sqrt{\lambda_i}, \sqrt{\lambda_i}]$ do often look dissimilar to the training shapes. For example, in Figure 5.6, the shapes shown in Figures 5.6(a) and 5.6(e) can not be considered as plausible instances of lumbar vertebrae. Thus, SCDMs are less specific than both the standard ASM, which excludes those shapes completely, and the Probabilistic ASM, which prevents them indirectly by assigning a higher shape energy to them. The limited specificity may especially become a problem in cases where the image information is very ambiguous such that there is a higher risk that it misguides the segmentation process.

SCDMs assign positive energies to shapes that leave the model's subspace. Thus, SCDMs penalize the DFFS, but in a regularized way: As can be seen from Equation 2.36, the internal energy of SCDMs splits the residual vector $\boldsymbol{r} \in \mathbb{R}^{3n}$ into n vectors $\boldsymbol{r}^i$, $\boldsymbol{r}^i \in \mathbb{R}^3$. These $\boldsymbol{r}^i$ can be visualized as a displacement field on the landmarks of model representation $\boldsymbol{y}(\boldsymbol{b})$ of $\hat{\boldsymbol{x}}$, and the internal energy enforces that this displacement field is smooth by penalizing differences between the $\boldsymbol{r}^i$ of neighboring landmarks. Thus, the resulting segmentation is smooth if $\boldsymbol{y}$ is smooth.

Note that this form of regularization is not possible for nonlinear shape priors based on Kernel PCA. The reason for this is that in Kernel PCA, the splitting of the shape into a component inside and a component outside a feature space is not made in the original input space $\mathbb{R}^{3n}$, but in the space induced by the chosen kernel. Because it is in general not possible to compute the preimage for the kernel principal components, one can not separate the actual shape into components that belong to the Kernel PCA's feature space and those that lie outside this feature space. This also means that one can not decompose the DFFS into a vector field with a intuitive geometric meaning as in the case of linear shape models. In contrast to the regularized DFFS used by SCDMs, the local energy of the Probabilistic ASM can be used for both linear and nonlinear models.

The effort required for parameter tuning is similar in both approaches, as both require a parameter that balances image and shape energy.

5.6.3. Nonlinear Active Shape Models

There is a conceptual difference between the Probabilistic ASM and previously proposed nonlinear ASMs: The Probabilistic ASM tries to find a compromise between image information and prior knowledge. In contrast to that, the nonlinear ASMs of Romdhani et al. [RGP99], Cootes and Taylor [CT97] and Twining and Taylor [TT01] follow the idea of a 'space of plausible shapes' used in the standard ASM, in which a shape is either considered as plausible or not. It turns out that the realization of an allowable shape space for nonlinear ASMs is nontrivial, especially for Kernel PCA.

The approach of Romdhani et al. [RGP99] has, beside other drawbacks, the fundamental problem that the ill-defined preimage problem must be solved. Both Cootes and Taylor [CT97] and Twining and

Taylor [TT01] define shape energies to constrain shapes. However, rather than balancing this energy with image information, it is optimized until the allowable shape space is reached. However, it is not obvious how this threshold that splits between plausible and implausible shapes can be adequately defined. A second disadvantage is that the optimization projects implausible shapes to the isosurface defined by the threshold, but not further. Finally, there is the risk that the constrained shape bears no similarity with the deformed shape, because this similarity is not explicitly stated in the optimization problem.

The idea of finding a compromise between image information and prior knowledge, rather than defining a space of plausible shapes, allows the Probabilistic ASM a unified treatment of linear and nonlinear shape priors. The optimization problem boils down to computing a (possibly local) minimum of the combined energy. Because both linear and nonlinear shape priors are represented by an energy, changing the prior only means to plug in a different energy function into the framework; the algorithmic aspects remain the same.

5.6.4. Diffusion Snakes

The Probabilistic ASM uses the same linear and nonlinear shape energies that have been used previously in the 2D Diffusion Snakes of Cremers et al. [CTWS02, CKS03] (see also Section [CTWS02]). These shape energies can in turn be regarded as direct applications of the more general concept of splitting log-likelihoods into DIFS and DFFS as proposed by Moghaddam and Pentland [MP97].

Although the shape energies are identical, the segmentation approach is completely different: Diffusion Snakes evolve a segmentation contour by minimizing the energy functional using gradient descent until a local minimum is reached. This local minimum is the final segmentation result. In contrast to that, the Probabilistic ASM uses the typical EM-like adaption approach taken by all ASM variants, which alternates between selecting a sparse set of image features in the neighborhood of the current model boundary, and imposing shape constraints. The energy is here minimized several times, once in each iteration, but different image features are used in each iteration. Neither this work, nor the work of Cremers et al. provide a comparative evaluation of both segmentation strategies, such that it can not be stated which approach works better. Because Cremers et al. do not provide quantitative results on segmentation accuracy or computation time, even an indirect comparison is difficult. Given the fact that contour evolution is a strategy often employed when using implicit shape priors, and these algorithms do not outperform state-of-the-art ASM variants both in accuracy and computation time, it seems unlikely that a contour evolution strategy is in general superior to the EM-like approach of ASMs.

A second difference between the two algorithms is how pose parameters are handled: The Probabilistic ASM treats pose parameters as external parameters, like all other ASM variants. In the Diffusion snake framework, scale and rotation are explicitly optimized. While Cremers et al. [CTWS02] report this strategy to be advantageous over treating scale parameters as external parameters, it has already been explained in Section 2.5.2 that it also implies a hugh computational burden when computing gradients, especially in 3D. Furthermore, the experiments of Cremers et al. that suggest that optimizing pose leads to better results are only made for Diffusion Snakes, not for ASMs. It should be noted that very similar strategies to that used by the ASM are actually used for rigid registration of point sets: For example, the ICP [BM92] alternates between building point correspondence relations and computing optimal transformations.

The third difference is how smoothness is enforced: As already discussed in Section 5.3, such means are necessary because penalizing only DFFS allows arbitrary, even very jagged, segmentation contours. In Diffusion Snakes, the assumption of smoothness is no explicit part of the model. Albeit the Diffusion Snakes model has no smoothness term, the formulas that define the actual contour evolution in practice incorporate terms that explicitly penalize large distances between control points, which has a smoothing effect. In the Probabilistic ASM, the smoothing force is an explicit part of the model. Furthermore, the use of a local shape model allows an adaptive smoothing that prevents that high curvature features of the shapes are smoothed away.

6. Applications

This chapter presents three applications of the Probabilistic Active Shape Model to medical image data: It is used to segment the liver and vertebrae in contrast enhanced CT scans, and for prostate segmentation in T2-weighted MRI scans. The goal of these applications is to provide a detailed evaluation of the algorithm described in Chapter 5.

The applicability of the Probabilistic ASM to a wide range of segmentation problems in medical imaging is shown by evaluating it on three organs with very different properties: While the liver and the prostate are soft tissue organs, vertebrae are bone structures. The prostate has a relatively simple, convex shape which does not vary that much, whereas the large anatomical variation of the liver is particularly challenging for shape modeling. Similarly, the vertebra shape varies significantly, depending on whether the organ is a thoracic or a lumbar vertebra.

In all three applications, the Probabilistic ASM is compared against the standard ASM. This comparison proves that the improved shape model of the Probabilistic ASM is not merely of theoretical value, but has a positive impact on the segmentation accuracy.

This chapter is structured as follows. In Section 6.1, a learning based approach for organ detection is presented. This algorithm is used for liver and prostate segmentation in order to locate the organ in the scan and to initialize the segmentation. Section 6.2 describes a segmentation pipeline for delineating the liver in contrast enhanced CT scans. Section 6.3 presents the vertebra segmentation algorithm. Finally, Section 6.4 deals with the prostate segmentation framework.

6.1. Organ detection

Like the standard ASM, the Probabilistic ASM is a local search algorithm and therefore requires a suitable initialization which places the shape model roughly onto the target structure in the image. For liver and prostate segmentation, an automatic organ detection algorithm is used which provides this initialization. This algorithm is described in the following section.

As worked out in Section 2.4.2, machine learning-based approaches for organ detection have several advantages: They provide robust and fast detection rates, and are at the same time versatile because they can be trained for the localization of a huge variety of different organs. The algorithm presented here is based on the Viola-Jones face detection framework [VJ01]. Most parts of it have been developed by Florian Jung during his master's thesis [Jun12]. Additionally to his framework, a pose correction algorithm for CT scans has been developed in this work.

The organ detection approach can be decomposed into a training and a detection phase.

6.1.1. Training phase

In the training phase, the detector is learned from a set of training images. In order to train the detector for a particular organ, it is assumed that the bounding box of the organ in each training image is annotated. From each training image, one *positive* and several *negative* training examples are extracted. Each example corresponds to a subimage of the original scan, resampled with a fixed resolution. The subimage defined by the organ's bounding box is used as the positive example. The negative examples are obtained by randomly sampling subimages that have roughly the same size as the positive example, but have no or only a small overlap with the bounding box.

The detector is learned from the examples by selecting the most discriminative 3D Haar-like features within the detector window using AdaBoost. As proposed by Viola and Jones [VJ01], decision tree stumps are used as weak classifiers. The value of a Haar-like feature is the difference between the sum of intensities of adjacent rectangular image regions (see Figure 6.1(a)). The advantage of Haar-like features is that they can be quickly evaluated using integral images. For the organ detector, seven distinct 3D Haar-like feature types proposed by Cui et al. [CLS*07] are used. Additionally, an *intensity sum* feature is added, whose feature value is simply the sum of intensity values within the feature region. The eight different feature types used in this thesis are visualized in Figure 6.1(b). The set of different features of a particular feature type is usually very huge, because the features can differ in size and can have different relative positions within the detector window.

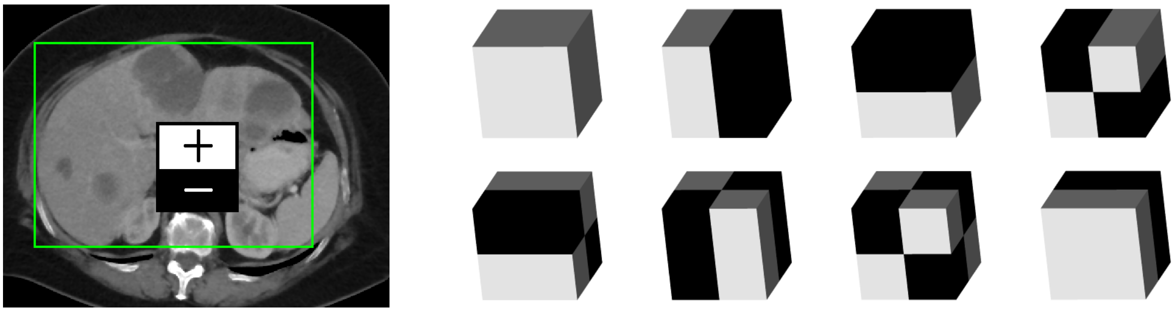
A bootstrapping approach similar to that of Sung et al. [SP98] can optionally be used to increase the robustness of the classifier: After the detector has been trained, it is used to detect the organ in each training image. False positives of the detector are added as negative examples to the training set, and the detector is retrained.

The training phase is relatively time consuming. Training takes from several hours up to days or even weeks, depending on the selected resolution of the extracted examples. The reason for this complexity is that the training phase performs an exhaustive search over the set of all Haar-like features in a subimage. This set is an overcomplete description of the subimage and its size is by far greater than the number of voxels.

6.1.2. Detection phase

The organ is detected in a tomographic image using the *sliding window* approach which exhaustively searches for the organ in each subimage of a particular size. For each subimage, the 3D Haar-features selected during the training phase are evaluated and passed to the trained detector which decides whether the subimage contains the organ or not. In order to account for size variability of the organ, the detection is repeated several times with detection windows of varying physical size, which is technically achieved by resampling the input image with different spacings.

The detector usually reports not only a single, but several detections of the sought organ. In most cases, the detections are clustered in the organ's vicinity and can be regarded as correct classification, but outliers also occur. Because the detection with the highest score does not necessarily correspond to



(a) Calculation of Haar-like features

(b) Intensity feature and 3D Haar-like features

Figure 6.1.: (a): The value of a Haar-like feature is the difference between the sum of intensities of adjacent rectangular image regions. In this example, the sum of intensities of the black region is subtracted from the sum of intensities of the white region. (b): Features used in the object detector. The top-left feature is a pure intensity-sum feature, all other seven features are 3D Haar-like features.

a correct classification, a cluster-based approach is used to account for outliers. The mean position of the first k detection results is computed, and those results with a high distance to the mean are discarded as outliers. From the remaining detections, the result with the highest score is chosen as the detected bounding box.

The organ detector is very fast: Even if the image is scanned several times with varying resolutions, it requires usually only one or two seconds, depending on the size of the image.

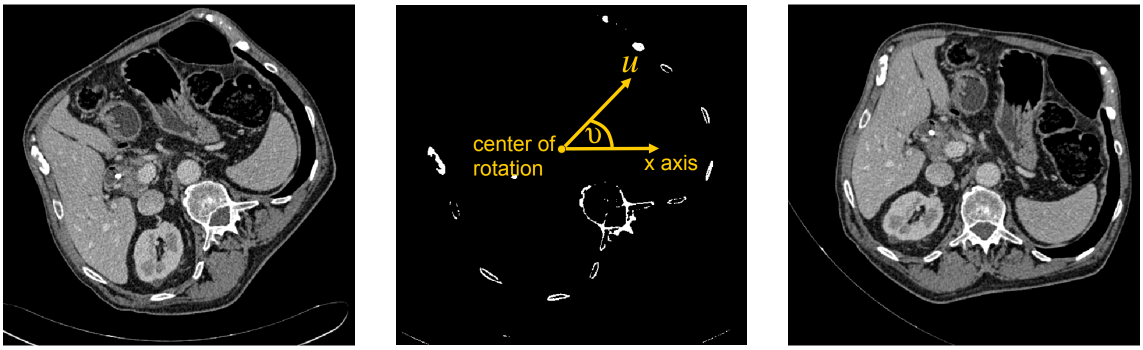
6.1.3. Extensions

In order to increase the robustness of the detection, three additional techniques can be optionally used in order to improve the detection accuracy in particular applications: Patient pose correction, enlargement of the region of interest (ROI), and multi-organ detection.

6.1.3.1. Patient pose correction

The detection algorithm assumes that all patients have the same pose. In particular, it assumes that they lie upright and on their back in the scanner. This assumption is in most cases true due to the standardized CT scanning protocol. However, there are some rare cases in which the patients lie on their side, as shown in Figure 6.2(a). Because the detection algorithm is not able to directly handle such scans, an algorithm has been developed which automatically identifies scans in which patients lie on their side, and rotates them accordingly.

The pose correction algorithm assumes that the input scan is a CT image that captures the abdominal region, that the patient's rotation is restricted to the xy plane, and that the patient's rotation is moderate. The basic idea is to infer the patient's pose by estimating the orientation of his skeleton, which can



(a) Original scan (b) Thresholded scan (c) Rotated scan

Figure 6.2.: Illustration of automatic pose correction of CT scans. (a): Original scan. (b): Thresholded scan which is used to estimate the patient's pose. The vector $\boldsymbol{u}$ is the first principal axis of the thresholded bone tissue. (c): Corrected scan obtained by rotating the scan by $-\upsilon$ around the estimated center of gravity of the patient.

be robustly extracted from CT scans. A thresholding is used to extract bone voxels from the scan. All voxels with a HU value greater than 300 are considered as bone voxels. A PCA of the x and y components of the coordinates of all bone voxels yields the first principal axis $\boldsymbol{u}$, which is the axis on which bone tissue has the largest spatial variance. It is assumed that $\boldsymbol{u}$ is orthogonal to the patient's sagittal plane and tangential to his coronal plane. It is further assumed that the angle υ between $\boldsymbol{u}$ and the positive x-axis is between $-90°$ to $90°$. If this is not the case, that is $u_1 < 0$, the direction of $\boldsymbol{u}$ is inverted by multiplying -1.

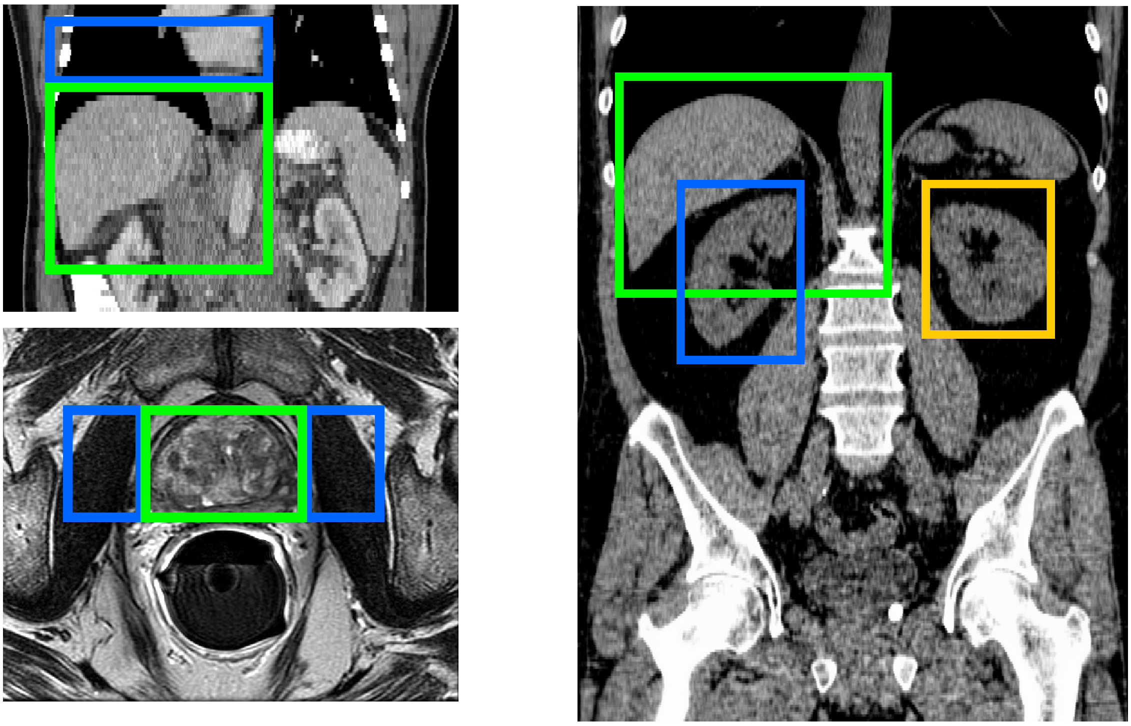
A pose correction of the image is only done if υ is outside the interval $[-15°, 15°]$. In this case, a transformed image is computed that rotates the original scan. To this end, a rotation matrix $\boldsymbol{R}$ is computed that maps $\boldsymbol{u}$ to the positive x-axis:

$$R = \begin{pmatrix} \cos\upsilon & -\sin\upsilon & 0 \\ \sin\upsilon & \cos\upsilon & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.1}$$

where

$$\upsilon = \arctan\left(\frac{u_2}{u_1}\right). \tag{6.2}$$

As center of rotation, the patient's center of gravity is used, which is estimated by computing the mean of all voxel coordinates with an HU value greater than -900. This effectively excludes voxels belonging to the surrounding air. An illustration of the pose correction algorithm is shown in Figure 6.2.



(a) Enlarging the ROI

(b) Multi-organ detection

Figure 6.3.: The robustness of organ detectors can be increased by exploiting contextual information. (a): By enlarging the ROI, regions with high contrast to the sought organ are included into the ROI, such that more discriminative Haar-like features are found. (b): Multi-organ detection allows to identify outliers of individual detectors through anatomical constraints modeled by a global location model.

6.1.3.2. Enlarging the region of interest

In contrast to faces, many organs have a relatively homogeneous texture such that Haar-like features within the organ have only weak discriminative power. On the other hand, the surroundings of organs are not arbitrary, but restricted due to anatomical constraints. It is therefore often beneficial to enlarge the ROI such that the detection is no longer restricted to features within the bounding box of the organ. To which extent and in which direction the ROI is enlarged depends on the application. In general, it is reasonable to include regions which have a high contrast to the sought organ. Examples for enlarging the ROI are illustrated in Figure 6.3(a). The enlargement can be reversed after detection in order to obtain the organ's bounding box. A drawback of enlarging the ROI is that not only the organ, but also the surroundings must be completely visible in the scan. This restriction does not only affect the training images, but also the scans to which the detector is applied.

6.1.3.3. Multi-organ detection

The idea of multi-organ detection is to combine several single organ detectors with a global model that describes geometric relationships between organs. These additional anatomical constraints are used to identify misdetections, because misdetections are likely to be inconsistent with the result of other detectors. A PCA of the centers of the bounding boxes of multiple organs is used to learn a global location model. In the detection phase, each scan is separately processed with the individual organ detectors. A globally consistent detection of all organs is determined by an exhaustive search over all combinations of the best k results of the individual detectors. The combination is selected that achieves the best compromise between the quality of the individual detectors and the Mahalanobis Distance to the mean of the global location model. Similar to enlarging the ROI, the approach has the disadvantage that all organs must be visible in the scan. Multi-organ detection of liver and both kidneys is illustrated in Figure 6.3(b).

6.2. Liver segmentation

Liver segmentation in contrast enhanced CT scans plays a crucial role in surgery planning. It is required, for example, for pre-operative planning of tumor resections [MTC02]. In this application, the visualization of patient specific anatomy requires labeling of relevant structures, such as the liver, liver tumors and the hepatic vessel tree.

Heimann et al. [HvGS*09] point out several challenges that have to be met when segmenting the liver in contrast enhanced CT scans: Low contrast to its surroundings, a high shape variability and the presence of pathological anatomy caused by tumors or by surgical interventions such as resection. The low contrast of the liver to adjacent structures such as stomach or heart renders purely intensity or region-based segmentation impossible. This is why approaches for automatic liver segmentation are usually model-based and rely to a great extent on prior knowledge. But the high shape variability and the presence of pathological anatomy makes even model-based segmentation particularly difficult, because the models need to be robust and flexible enough to handle these issues.

Most automatic state-of-the-art algorithms for liver segmentation are shape model-based. They use either level set segmentation with implicit surface representation [WSH09] or some variant of the ASM [KLL07, HMMW07, EKSW10]. The latter are hybrid approaches, because they do not rely completely on learned shape variation, but also incorporate additional model assumptions that often depend on one or more free parameters. The common goal of these additional models is to relax the learned shape constraints in order to achieve more accurate segmentations.

This section presents an application of the Probabilistic ASM to liver segmentation in contrast enhanced CT scans. As opposed to the related approaches mentioned above, no additional models are used for modeling shape variation. Both global and local shape models are completely learned from the training examples.

6.2.1. Overview

For segmenting the liver in contrast enhanced CT scans, an automatic segmentation pipeline has been devised. It consists of the following steps:

1. **Preprocessing.** The input image is smoothed with a median filter. The intensity of a voxel is replaced by the median of the intensity values of its 26-neighborhood. Preprocessing is done for increasing the robustness in case of noisy input images.
2. **Detection.** The detection algorithm presented in Section 6.1 is used for detecting the liver. Details of the configuration of the detector are given in Section 6.2.2.
3. **Intensity estimation.** Important parameters that describe the intensity distribution of liver tissue are extracted from the scan. Intensity estimation is done using an approximate segmentation, or *pre-segmentation*, which is derived by rastering the mean shape into the detected bounding box. The algorithm for intensity estimation is described in detail in Section 6.2.3.
4. **Pose estimation.** Rotation, scale and liver shape are estimated using a modified ICP. This step is explained in Section 6.2.4.
5. **Intensity re-estimation.** The liver intensity estimation is repeated in order to refine the estimates. Because the pre-segmentation after pose estimation is more accurate than after liver detection, a more confident intensity estimation is possible.
6. **Probabilistic ASM.** The final segmentation is computed using the Probabilistic ASM. Details of the parameters of the algorithm and about the appearance model used for boundary detection are described in Section 6.2.5.

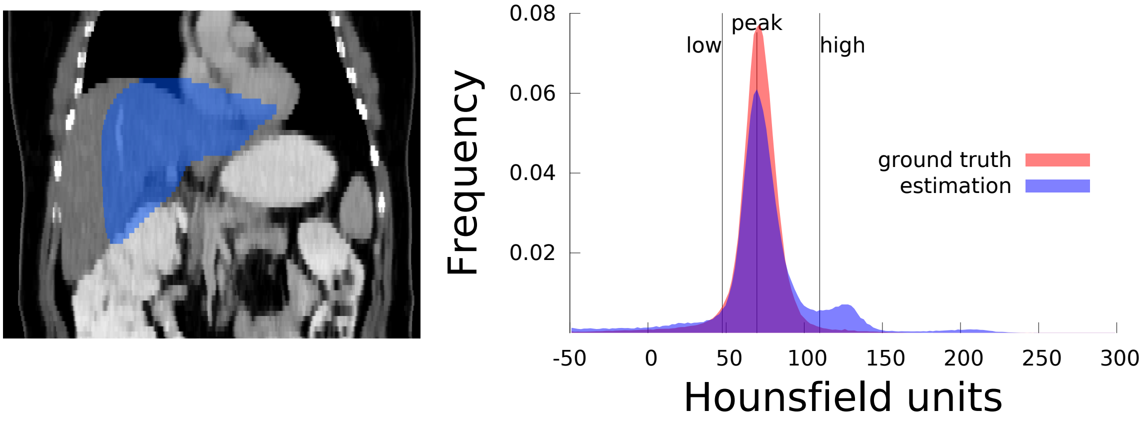
6.2.2. Detection

The liver is automatically detected in the CT scan using the organ detection algorithm presented in Section 6.1. In order to increase the robustness, the ROI that is detected is not just the liver's axes-aligned bounding box, but a slightly larger region: To this end, length and width of the training examples, which are originally the bounding boxes of the expert segmentations of the liver scans, are scaled by 10%. Even though the liver has a very high contrast to the lung, it has been decided not to scale the height of the boxes, because some CT scans have a very restricted field of view such that only very small parts of the lungs are visible.

The size of the window used for training and detection has a resolution of $22 \times 20 \times 15$ voxels. In order to handle CT scans in which the patient lies askew, the detection algorithm uses the patient pose correction algorithm presented in Section 6.1.3.1.

6.2.3. Intensity estimation

After having detected a region of interest which contains the liver, the intensity interval of liver tissue is estimated. Even though intensities in CT imaging are standardized through the Hounsfield scale, the



(a) Pre-segmentation (b) Histogram

Figure 6.4.: Example of liver intensity estimation. (a): A pre-segmentation, obtained by rastering the mean shape into the detected bounding box, is used to compute a histogram. (b): This histogram (blue area) is analyzed in order to obtain the parameters I_{low}, I_{high} and I_{peak}. The red area is the histogram of the expert segmentation.

actual range of Hounsfield units (HU) to which liver tissue is mapped varies significantly from image to image. One reason for this variation is the use of contrast agents for emphasizing liver tissue: For example, the time span between contrast agent injection and image acquisition greatly influences how bright the liver appears in the image.

The intensity of liver tissue is modeled by a peak intensity I_{peak} and an interval $[I_{\text{low}}, I_{\text{high}}]$. The interval bounds must be determined in such a way that they include most liver voxels, but exclude most other tissue types. The basic idea is to create a binary mask by rastering the mean shape into the detected liver's bounding box, like in Figure 6.4(a), and to determine the parameters by an histogram analysis of the masked image region, as shown in Figure 6.4(b). The histogram is restricted to voxels within the HU interval [-50, 300]; this intensity interval is chosen wide enough to include virtually all liver voxels in different contrast enhanced CT scans.

The following algorithm is used to determine I_{low}, I_{high} and I_{peak} from the histogram: The peak intensity I_{peak} is set to the HU value with the highest bin frequency within the histogram. Then, starting with $I_{\text{low}} = I_{\text{high}} = I_{\text{peak}}$, the interval $[I_{\text{low}}, I_{\text{high}}]$ is gradually increased: If the frequency of the histogram bin adjacent to I_{high} is larger than the frequency of the bin adjacent to I_{low}, I_{high} is increased by one bin spacing, if not, I_{low} is decreased by one bin spacing. The algorithm terminates if the sum of bin frequencies within $[I_{\text{low}}, I_{\text{high}}]$ is larger than 98 %. Moreover, a second rule is required to stop that the interval becomes too big if the histogram is noisy, that is, in cases where the voxels within the masked image region are not exclusively liver voxels. To this end, the interval bound are kept fixed if their distance to I_{peak} is larger than 30 HU, and the frequency of the nearest bin outside the interval is smaller than 10 % of the frequency of the bin at I_{peak}.

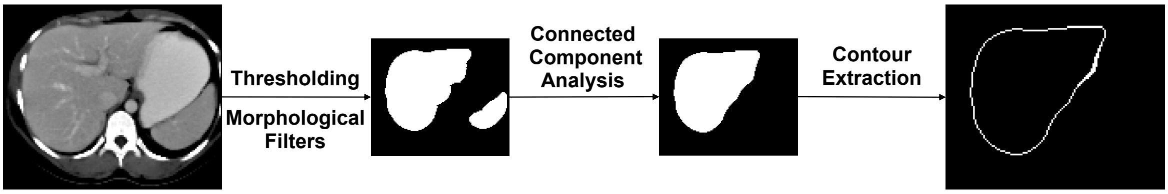


Figure 6.5.: Filtering pipeline for extracting voxels that lie on the liver's boundary.

6.2.4. Pose estimation

Preliminary experiments have shown that ASMs often do not modify the initial estimate of the organ's orientation significantly during execution. The orientation of the liver has a quite high variation which is independent from the patient's pose. Therefore, it is important to provide a good initial estimation of the liver's orientation in order to achieve accurate segmentation results. In this work, the liver's orientation is estimated by registering the shape model against a set of image features using a modified ICP algorithm. In contrast to the original ICP proposed by Besl and McKay [BM92], this algorithm does not only estimate translation and orientation, but also scale and shape.

The image features against which the shape model is registered are boundary voxels that lie within the estimated liver intensity range. They are extracted using the image filter pipeline shown in Figure 6.5, which computes a binary image. The original image is filtered using thresholding so as to extract voxels within the estimated liver intensity interval $[I_{\text{low}}, I_{\text{high}}]$. This image is refined with two morphological filters, opening and hole filling, in order to disconnect different structures and remove small islands within the segmented region. Then, a connected component analysis is done, and the largest connected component of foreground voxel is identified. It is assumed that this component corresponds to the liver, because the liver is the largest organ. Finally, a contour filter is applied in order to identify boundary voxels. From these voxels, a subset of at most 50000 voxels is randomly sampled, against which the shape model is registered.

The registration of the shape model against the image features is done with a modified ICP which estimates translation, orientation, scale and shape parameters. Like in the standard ICP, this estimation is carried out iteratively, using a point-to-point relation that assigns to every *moving point* its currently closest *model point*. In this application, the set of moving points are the landmarks of the shape model, and the set of fixed points are the image features. In the first iteration, the moving points are initialized with the mean shape $\bar{\boldsymbol{x}}$. Based on the closest-point relation to the image features, translation, scale and orientation are estimated using the method of Horn [Hor87]. However, instead of just computing a rigid transformation for $\bar{\boldsymbol{x}}$, each landmark is displaced to its currently closest point, which results in a deformed shape $\hat{\boldsymbol{x}}$. The standard procedure of ASMs described in Section 2.4.4.1 is applied to estimate the shape parameters and to compute the nearest 'plausible' shape $\boldsymbol{x}$. This plausible shape $\boldsymbol{x}$ constitutes the set of moving points in the next iteration. The modified ICP terminates after 40 iterations.

Like the standard ICP, the modified ICP must itself be initialized with estimates of pose parameters that are used in the first iteration. Estimates of the scale factor and the translation are derived from the detected liver bounding box. The identity matrix is a reasonable initial choice for the rotation matrix, because it corresponds to the 'mean' liver pose of the training shapes. However, like the original ICP algorithm, the modified ICP algorithm can get trapped in local optima. Therefore, the robustness is additionally increased by executing the modified ICP several times, using different initial scaling factors and rotation matrices. As scale factors, $0.8q$, $0.9q$, q, $1.1q$ and $1.2q$ are used, where q is the smallest side length of the detected bounding box. Likewise, two matrices are tested that describe a rotation around the y-axis, where the rotation angle υ is set successively to the values $-20°$ and $0°$. Combining the five scale factors and two rotations amounts to ten different initializations. In order to choose the final estimation of pose and shape parameters from the ten registration result, the result $\boldsymbol{x}^*$ is chosen which minimizes the function

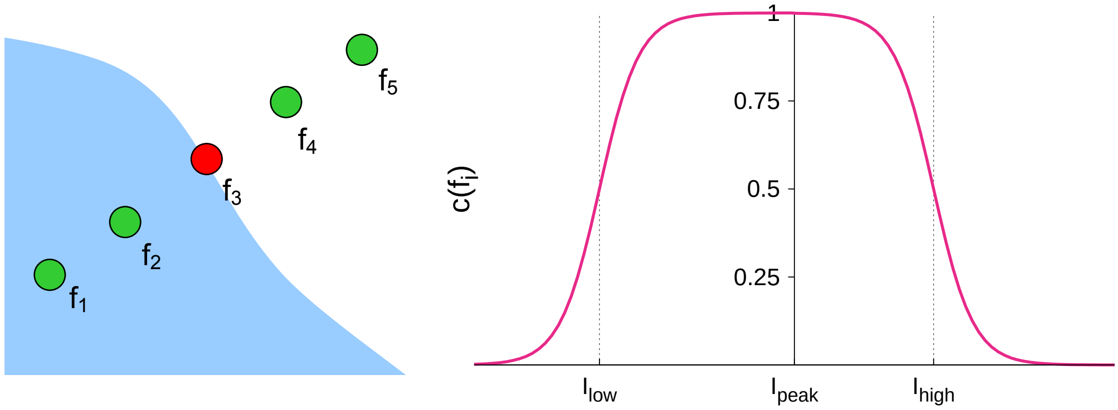
$$h(\boldsymbol{x}) = \sum_{i=1}^{n} \|\boldsymbol{x}^i - C(\boldsymbol{x}^i)\|^2 \tag{6.3}$$

where $C(\boldsymbol{x}^i)$ denotes the nearest image feature to landmark $\boldsymbol{x}^i$. $h(\boldsymbol{x})$ measures how well the shape $\boldsymbol{x}$ matches the image features.

6.2.5. Probabilistic ASM

The final segmentation is done with the Probabilistic ASM. In each iteration, image features are sampled at nine candidate positions around each landmarks. All candidates lie on a line which is centered at the current landmark's location, and which is oriented along the landmark's normal. The spacing between two candidates is 1 mm. In order to deform the shape, one candidate must be selected for each landmark. For liver segmentation, candidate selection is done using three components of the algorithm, where each component works on a different layer of abstraction:

1. The *appearance model* assigns a numerical value, or *fitness*, to an image feature. The appearance model considers only one candidate at a time. It is described in Section 6.2.5.1.
2. *Rule-based boundary search* is used for guiding the image feature search in cases in which none of the nine candidates of a landmark is considered as a boundary feature. In this case, the fitness values assigned by the appearance model are overwritten in order to facilitate either growing or shrinking of the shape. Rule-based boundary search, which is explained in detail in Section 6.2.5.2, considers all candidates of one landmark.
3. *Optimal surface detection* [HMMW07] (see Section 2.4.3.3) is used in order to determine a consistent set of candidate features, one for each landmark, and thereby decides how the shape is deformed. The smoothness parameter Δ is set to 2. However, if the angle between the surface normals of two neighboring landmarks exceeds 15° on the mean mesh, the smoothness parameter is set to the number of candidates, that is to 9, for these landmarks. This effectively switches off the smoothness constraints at high curvature features and therefore enables a higher segmentation accuracy of such details. Optimal surface detection considers all candidates of all landmarks.



(a) One-dimensional profile

(b) Classification function

Figure 6.6.: (a): Illustration of a one-dimensional profile, oriented along the surface normal. In this example, a five-dimensional feature vector $\boldsymbol{f}$ is created by sampling intensity values at the positions marked by the green and red circles. (b): The graph shows a plot of the function $c(f_i)$ for classifying an intensity value f_i into the classes *liver* and *background* using the estimated liver intensity interval $[I_{\text{low}}, I_{\text{high}}]$ and the liver's peak intensity I_{peak}.

When the shape has been deformed, shape constraints are imposed as described in Chapter 5. The Probabilistic ASM terminates after a fixed number of 60 iterations.

6.2.5.1. Appearance Model

In order to detect the liver boundary during segmentation, feature vectors are extracted from the image at each candidate position of a landmark. A feature vector $\boldsymbol{f} \in \mathbb{R}^l$ consists of l intensity values, which are sampled along a one-dimensional linear profile. The profile is oriented along the landmark's surface normal, as shown in Figure 6.6(a). The task of the appearance model is to assess the feature vectors of all candidates in order to determine the landmark's next position.

In his Master's thesis [Fre12], Sebastian Freutel evaluated several fitness functions that assess the quality of feature vectors in context of liver segmentation. He obtained the best segmentation results when considering the feature vector entries f_i independently, rather than learning a multivariate distribution over $\boldsymbol{f}$. The appearance model that performed best in in his experiments is based on the idea of classifying each f_i independently into the classes *liver* and *background* using a sigmoid function. In this work, an extension of this fitness function is used. It incorporates several improvements of the original function proposed by Sebastian Freutel, such as emphasizing a high gradient at the profile's center, and using different sigmoids for classifying low and high intensity values such that skew intensity distributions can be better modeled.

A profile $\boldsymbol{f} \in \mathbb{R}^l$ of length l, where l is an odd number, is assessed using the fitness function

$$g(\boldsymbol{f}) = g'(\boldsymbol{f}) + (1 - 2 \cdot |c(f_{l^*}) - \frac{1}{2}|) + \sum_{i<l^*} c(f_i) + \sum_{i>l^*} (1 - c(f_i)) \tag{6.4}$$

where $l^* = \lceil \frac{l}{2} \rceil$,

$$g'(\boldsymbol{f}) = \begin{cases} \dfrac{2|f_{l^*+1} - f_{l^*-1}|}{\nabla_{\max}} & |f_{l^*+1} - f_{l^*-1}| \leq \nabla_{\max} \\ 2 & \text{otherwise} \end{cases} \tag{6.5}$$

and

$$c(f_i) = \begin{cases} \dfrac{1}{1 + \exp(-\sigma(f_i - I_{\text{low}}))} & f_i \leq I_{\text{peak}} \\ 1 - \dfrac{1}{1 + \exp\left(-\sigma(f_i - I_{\text{high}})\right)} & f_i > I_{\text{peak}}. \end{cases} \tag{6.6}$$

It is assumed here that f_{l^*} is sampled at the candidate's position, the entries f_i with $i < l^*$ are sampled in negative normal direction, and the entries f_i with $i > l^*$ are sampled in positive normal direction. This ordering is also visualized in Figure 6.6(a).

The function $c(f_i)$, which is plotted in Figure 6.6(b), makes a soft classification of a sampled intensity value into the classes *liver* and *background*. Intensities near I_{peak} are mapped approximately to one, intensities above I_{high} or below I_{low} approximately to zero. If $f_i = I_{\text{low}}$ or $f_i = I_{\text{high}}$, it is $c(f_i) = 0.5$. The function $g'(\boldsymbol{f})$ is used to emphasize high gradients at the profile's center.

Finally, the function $g(\boldsymbol{f})$ fuses the information of $g'(\boldsymbol{f})$ and $c(f_i)$, $i = 1, \ldots, l$. Intuitively, $g(\boldsymbol{f})$ is an edge detector which searches for a transition from liver tissue to background. Preliminary experiments showed that using a relatively long profile length increases the robustness of the detector. In the experiments below, $l = 9$ is chosen. The threshold $\nabla_{\max}$ for clamping the gradient magnitudes is set to 10, and the scale factor of the sigmoid σ to 2. Furthermore, nearest neighbor interpolation is used when sampling $\boldsymbol{f}$, which performed significantly better in experiments than linear interpolation.

6.2.5.2. Rule-based boundary detection

Rule-based boundary detection is a method proposed by Erdt et al. [EKSW10] for enlarging the capture range of an ASM by guiding the model towards the organ boundary. The basic idea is to identify situations in which a landmark is so far away from the organ boundary that no candidate lies on the boundary. Two basic cases can be distinguished: The landmark is either inside, or outside the organ. In the first case, rule-based boundary detection moves the landmark to the most distant candidate in positive normal direction, and in the second case, to the most distant candidate in negative normal direction.

In this work, the decision of whether such rules apply is done by classifying image features $\boldsymbol{f}$ as being *inside* or *outside* the liver. A feature $\boldsymbol{f}$ is assumed to be inside if $c(f_i) \geq 0.5$ for $i = 1, \ldots, l^* + 1$, and outside if $c(f_i) < 0.5$ for $i = 1, \ldots, l^* - 1$. If all features $\boldsymbol{f}$ are classified as being inside, the fitness

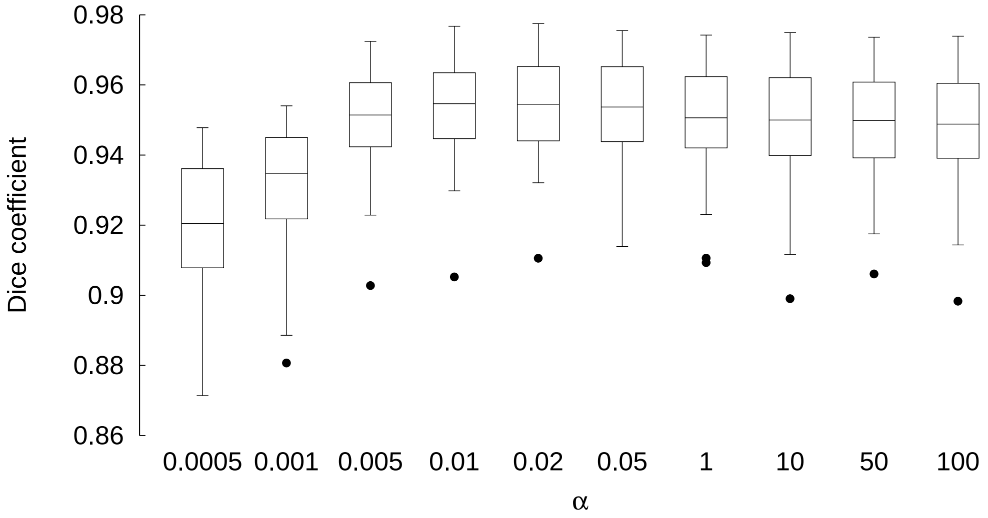


Figure 6.7.: Dice coefficients achieved on LOWRES and SLIVER07-TRAIN using different choices for α.

of the most distant candidate feature in positive normal direction is set to $\frac{l+1}{2}$, and the fitness of all other candidates is set to 0. Likewise, if each sampled intensity value at each candidate is outside the estimated liver intensity interval, the fitness of the most distant candidate feature in negative normal direction is set to $\frac{l+1}{2}$, and the fitness of the remaining landmarks is set to 0. Note that the maximal fitness value assigned by rule based boundary detection is approximately a half as large as the highest fitness values assigned by the appearance model, because it is

$$l+1 = \sup_{\boldsymbol{f} \in \mathbb{R}^l} g(\boldsymbol{f}). \tag{6.7}$$

Further note that in the implementation made in this work, rule-based boundary detection only modifies the fitness values of candidates in some special cases. Which of the candidates is actually chosen as new position for a landmark is done by optimal surface detection.

6.2.6. Experiments

The 47 CT scans of the data sets LOWRES and SLIVER07-TRAIN (see Appendix B.1) are used for training the liver detector and the shape model, as well as for parameter determination of the components of the algorithm. In order to avoid testing on training data, cross-validation is used for all experiments reported on LOWRES and SLIVER07-TRAIN. 10-fold cross-validation is used for learning the detector, and leave-one-out cross-validation for shape model learning.

Correspondence is established using Parameter Space Propagation (Section 3.2.3) along a *JC-eps* shape similarity tree (Section 3.6.2). Curvature adaptive sampling (Section 3.4) is used to create landmark vectors with 3000 landmarks from each parameterization.

Table 6.1.: Average segmentation accuracy and standard deviation (SD) of the standard ASM, the linear Probabilistic ASM and the algorithm of Erdt et al. [EKSW10] on different liver data set. The reported running times for ASM and Probabilistic ASM have been measured using the hardware configuration C1 (see Appendix B.4). Erdt et al. [EKSW10] used a 2.93 GHz Quad Core CPU. Individual segmentation results of the Probabilistic ASM on the ten scans of Sliver07-Test are available on the website of the Grand Challenge (www.sliver07.org – Team *Grislies*).

	Dice	RVD [%]	ASD [mm]	RMS [mm]	HD [mm]	Time [s]
Standard ASM						
LowRes	0.92 SD: 0.02	-3.47 SD: 4.6	2.34 SD: 0.8	3.95 SD: 1.3	28.33 SD: 9.6	24 SD: 2.7
Sliver07-Train	0.93 SD: 0.01	-2.03 SD: 2.9	2.34 SD: 0.4	3.59 SD: 0.7	25.39 SD: 8.0	29 SD: 8.2
Sliver07-Test	0.92 SD: 0.02	-2.24 SD: 3.9	2.57 SD: 0.6	3.88 SD: 1.1	26.59 SD: 10.0	35 SD: 16.7
IRCAD	0.91 SD: 0.02	-4.15 SD: 5.1	3.06 SD: 1.1	5.39 SD: 2.9	36.83 SD: 17.0	22 SD: 3.8
Erdt et al.						
LowRes	0.93	-4.53 SD: 4.1	1.85 SD: 0.3	3.27 SD: 0.7	23.09 SD: 5.6	45
Sliver07-Train	0.96	1.28 SD: 2.2	1.30 SD: 0.3	2.67 SD: 1.0	26.52 SD: 10.1	
Sliver07-Test	0.95	1.32 SD: 2.0	1.54 SD: 0.4	3.13 SD: 1.0	25.90 SD: 7.3	
IRCAD	0.95	1.55 SD: 6.5	1.75 SD: 0.6	3.51 SD: 1.2	26.83 SD: 8.9	
Probabilistic ASM						
LowRes	0.95 SD: 0.01	-3.77 SD: 3.7	1.48 SD: 0.5	3.02 SD: 1.2	25.07 SD: 11.2	44 SD: 2.8
Sliver07-Train	0.97 SD: 0.01	-1.49 SD: 2.9	1.11 SD: 0.3	2.34 SD: 0.7	22.91 SD: 7.2	49 SD: 8.3
Sliver07-Test	0.96 SD: 0.01	-0.90 SD: 4.0	1.29 SD: 0.3	2.70 SD: 0.7	25.39 SD: 6.6	53 SD: 16.9
IRCAD	0.95 SD: 0.02	-3.62 SD: 5.5	1.94 SD: 1.1	4.47 SD: 3.3	34.60 SD: 17.7	41 SD: 3.8

The optimal parameter α for balancing shape and image energy is determined by trying a series of different values for segmenting LowRes and Sliver07-Train, and choosing the parameter that achieves the best average Dice coefficient in all experiments.

After having fixed all parameters, the automatic liver segmentation algorithm is evaluated on the 77 CT scans of the data sets LowRes, Sliver07-Train, Sliver07-Test and IRCAD. For comparison, these data sets are also segmented using the standard ASM. The same common parameters are used for the standard ASM and for the Probabilistic ASM, such that the approaches differ only in the way how shape constraints are imposed.

6.2.7. Results

The box plot in Figure 6.7 visualizes the segmentation accuracy, as measured with the Dice coefficient, of the proposed algorithm on LowRes and Sliver07-Train with different choices for the balancing parameter α. It is found that the best results are achieved with $\alpha = 0.02$, both visually as with respect to the average Dice coefficient. Therefore, α is set to 0.02 in all further experiments.

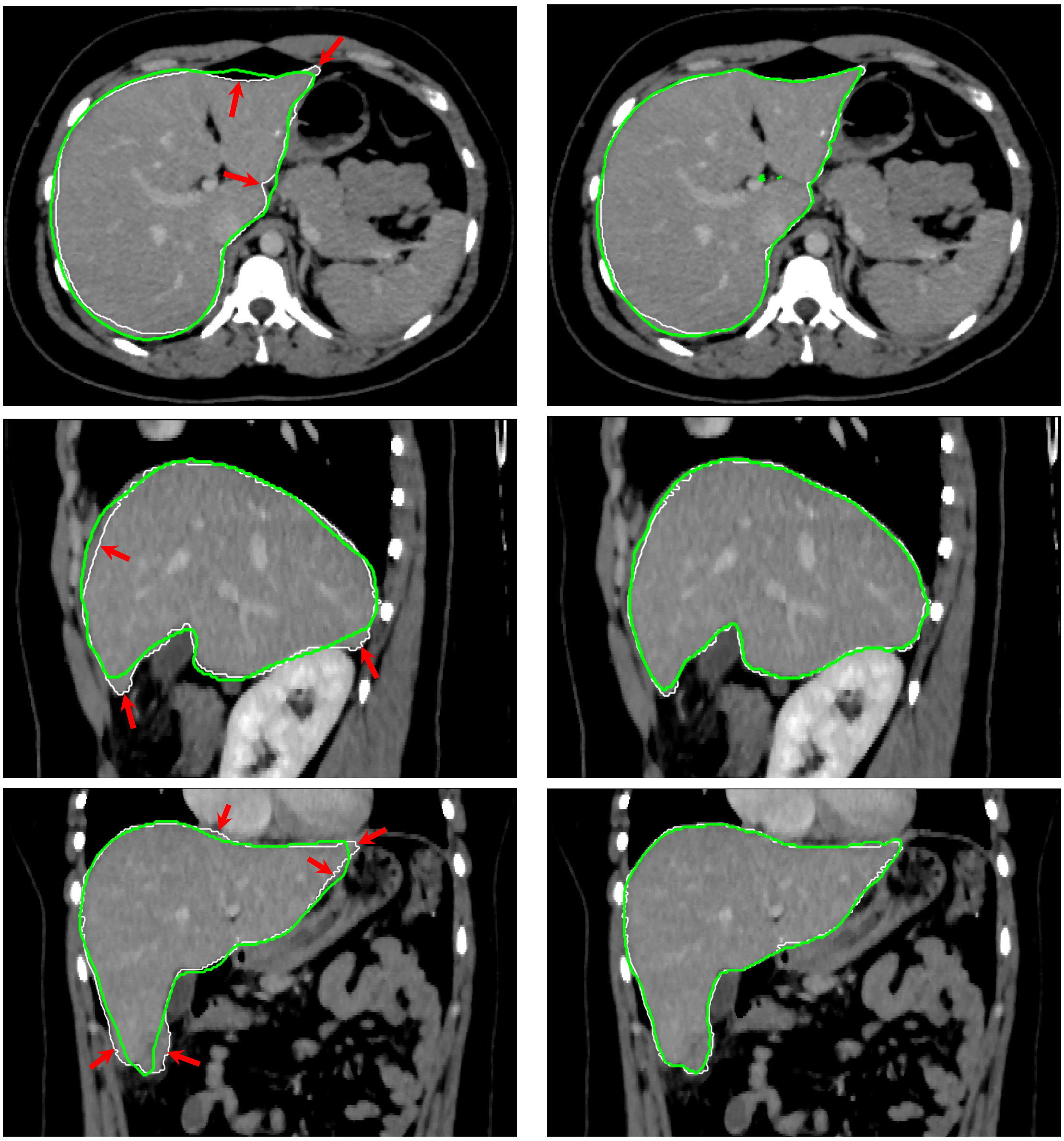


Figure 6.8.: Qualitative comparison of the segmentation of a scan from the IRCAD data set using the standard ASM (left column) and the Probabilistic ASM (right column). Shown are a transverse (top row), a sagittal (middle row) and a coronal slice (bottom row). The green contour shows the segmentation boundary, and the white contour the ground truth. Both segmentations have a high overlap with the ground truth segmentation, but the Probabilistic ASM delineates details more accurately. The red arrows in the left column mark regions in which the standard ASM is significantly less accurate than the Probabilistic ASM.

Table 6.1 shows the final segmentation results of the proposed liver segmentation algorithm achieved with different methods for constraining shapes. As an additional comparison, the table also lists results obtained by Erdt et al. [EKSW10], who tested their algorithm on the same data sets.

The Probabilistic ASM achieves considerably more accurate results than the standard ASM: For example, the ASD is more than 0.8 mm lower on LOWRES, more than 1.2 mm lower on SLIVER07-TRAIN and SLIVER07-TEST, and 1.1 mm lower on IRCAD. On the data set SLIVER07-TEST, the algorithm reaches a score of 69.9 in the scoring system of the Sliver07 Grand Challenge, while the ASM only reaches a score of 54.1. Please refer to Heimann et al. [HvGS*09] for details on score computation. In the challenge, the Probabilistic ASM reaches the sixth best score of over 20 different automatic segmentation algorithms (best score: 77.3)[1].

The Probabilistic ASM also performs slightly better than the algorithm of Erdt et al. [EKSW10] on all but the IRCAD data set. On the latter data set, the proposed algorithm reaches a lower accuracy than on all others. There is also a relatively high variance in the segmentation accuracy on IRCAD: For example, in three of the twenty cases, the ASD is higher than 3 mm.

A qualitative segmentation result that compares Probabilistic and standard ASM segmentation is shown in Figure 6.8. In this example, the segmentation contour computed by the Probabilistic ASM is closer to the ground truth, while the standard ASM segmentation smoothes some details away.

The time required for training the liver detector is approximately ten days using hardware configuration C3 (see Appendix B.4). The time for training the shape model is negligible. Having trained both the detector and the model, fully automatic liver segmentation takes in most cases less than a minute on a standard desktop computer (hardware configuration C1). In average, nine seconds are spent for median filtering. However, this step depends to a great extent on the image size: For the largest scan in the data sets, 36 seconds are required. Organ detection is done in one or two seconds in average. The actual time required for shape model adaption (steps 2. to 6. in Section 6.2.1) is approximately 37 seconds for the Probabilistic ASM and 17 seconds for the standard ASM.

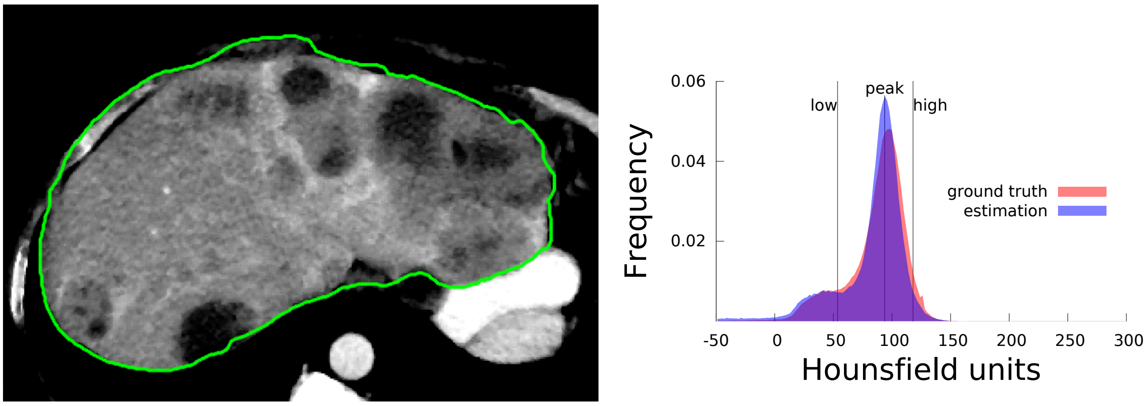
6.2.8. Discussion

The experimental results show that the Probabilistic ASM segments the liver considerably more accurately than the standard ASM. Both quantitative and qualitative evaluation results suggest that the reason for the higher accuracy is the increased flexibility of the Probabilistic ASM: Both algorithms achieve a comparable overlap, but the standard ASM is in many cases not able to delineate details accurately, like in the example in Figure 6.8. Because these details often exhibit a high contrast to the surroundings, the limited accuracy of the standard ASM must be caused by over-restrictive shape constraints. Because the shapes of segmentations computed by the Probabilistic ASM are not limited to linear combinations of the training shapes, it is able to include these regions into the segmentation.

The proposed liver segmentation algorithm achieves in total results that compare favorably against other state-of-the art algorithms for liver segmentation. The best results in the Sliver07 Grand Challenge

---

[1] As at the date of submitting the results (April 11, 2013)



(a) Example of oversegmentation

(b) Corresponding intensity interval

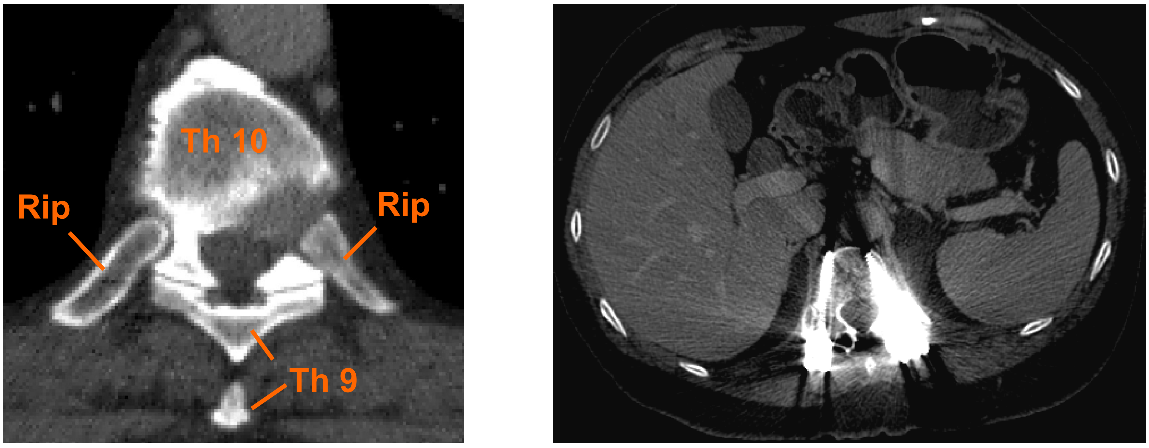
Figure 6.9.: In cases where the liver tissue is inhomogeneous due to large tumors, the estimated liver intensity interval (b) becomes rather large and therefore less specific. This may lead to an oversegmentation, as in the example in (a).

with an automatic approach have been achieved by Kainmueller et al. [KLL07], who reach in average a Dice coefficient of 0.97 and an ASD of 0.95 mm. The Probabilistic ASM comes close to these results, albeit it does not reach the same accuracy. Although Kainmueller's approach is also based on the Active Shape Model, they use a considerably more complex segmentation pipeline which consists of 14 different steps, including post-processing. Moreover, their shape model is learned from 102 examples, which means that twice as many training instances are used than in this work.

A reason for the relatively low segmentation accuracy on IRCAD is that this data set contains some very highly pathological cases with unusual shapes and large tumors. While the algorithm is capable of dealing with tumors of small or medium size, a large number of tumors or tumors of large size hamper intensity estimation. This issue is illustrated in Figure 6.9: Because tumors have lower intensity than healthy liver tissue, the estimated liver intensity interval becomes larger, and therefore less specific. In order to deal with this problem in future, one might explicitly incorporate a tumor intensity model into the algorithm, as it was done by Kainmueller et al. [KLL07].

In contrast to other shape model-based algorithms that have been proposed for liver segmentation [KLL07, HMMW07, EKSW10], the presented approach relies completely on learned global and local shape priors. Neither additional models, such as linear spring models [HMMW07], nor hybrid approaches [KLL07, EKSW10] that exchange the underlying model assumptions during segmentation are used.

The experiments show that such means are not required, because learned shape variability is sufficient to achieve a similarly high segmentation accuracy. Further improvements of the segmentation results are possible by improving the appearance model and the ASM's adaption strategy.



(a) Low contrast between adjacent bones

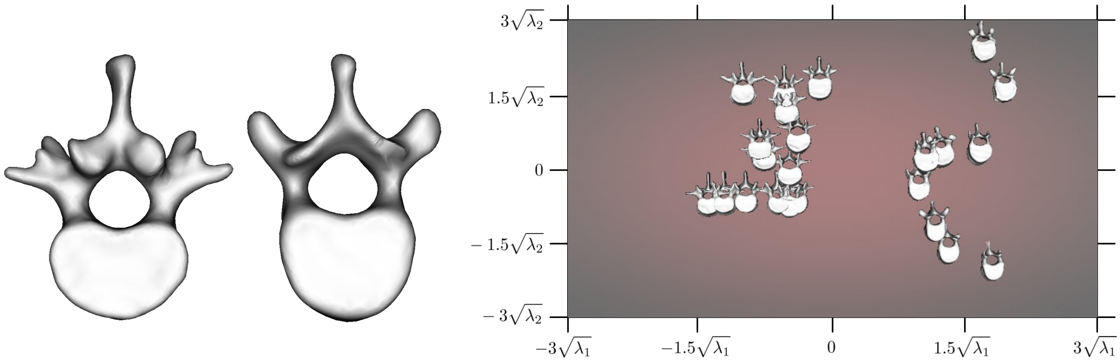
(b) Pedicle screws and metal shadows

Figure 6.10.: (a): Low contrast between adjacent vertebrae and between vertebrae and ribs makes purely intensity-based segmentations impossible. (b): Another issue that hampers accurate segmentation are pathologies or implants such as the pedicle screws. These implants also cause image acquisition artifacts—so-called 'metal shadows'— which are the thin rays emanating from the screw.

6.3. Vertebra segmentation

Segmentation of individual vertebrae in CT images is important in several applications: For example, segmentation is useful for supporting image-guided spinal biopsies, or for planning surgeries such as the insertion of pedicle screws [KOE*09]. Moreover, vertebra segmentation is necessary for computer aided diagnosis of osteoporosis [WKB*12]. Separating bone from other tissue is relatively easy in CT images, because bone tissue has a relatively high density and is therefore mapped to higher intensity values than soft tissue. However, the separation of a vertebra from neighboring vertebrae or other bones such as the ribs is nontrivial due to the low contrast between adjacent bones, as illustrated in Figure 6.10(a). Another problem that makes segmentation challenging is the presence of image artifacts or implants, such as the pedicle screws that can be seen in Figure 6.10(b). Because of these challenges, a model-based approach to vertebra segmentation is necessary.

A fully automatic, shape model-based approach for segmenting the complete spine has been proposed by Klinder et al. [KOE*09]. They use a separate shape model for each vertebra type, such that in total 24 different models are required: Seven models for cervical vertebrae (C1-C7), twelve for thoracic vertebrae (Th1-Th12) and five for lumbar vertebrae (L1-L5). Klinder et al. construct the models by averaging vertebrae of each type from ten training images. Unlike usual Statistical Shape Models, second order statistics are not considered in these mean shape models. In order to segment the spine, individual vertebrae are detected along the spinal curve. The type of a detected vertebra is inferred



(a) Lumbar and thoracic vertebrae

(b) Training shapes projected to first two principal axes

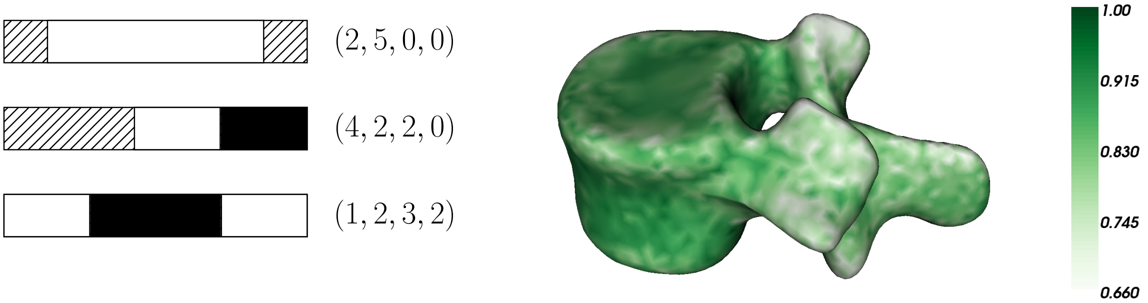
Figure 6.11.: (a): Exemplary shapes of lumbar (left) and thoracic (right) vertebrae. (b): When the 23 training shapes are projected to their first two principal components, two clusters can be observed. The left cluster contains all lumbar, and the right cluster all thoracic training shapes.

based on appearance features in order to select the correct shape model for segmentation. Finally, the selected shape model is adapted to the image using a variant of SCDMs (see Section 2.4.4.2).

In the following, an alternative approach for segmenting vertebrae is explored: A single statistical vertebra model is learned from 23 training shapes that include examples of three thoracic vertebra types (Th10-Th12) and three lumbar vertebra types (L1-L3). This shape model is used to segment vertebrae of the types L1-L3 and Th10-Th12 in test images. Because only one model is used, appearance-based identification of the vertebrae type is no longer necessary. However, the question arises whether a linear shape model learned from different vertebrae types is specific enough for accurate segmentation.

Especially lumbar and thoracic vertebrae have a distinct difference in shape, as illustrated in Figure 6.11(a). When the training shapes are projected to the first two principal components of their covariance matrix, as it is done in Figure 6.11(b), one can see that the shapes build two clusters, one containing lumbar, and the other thoracic vertebrae. If these shapes are used to learn a Gaussian probability distribution, that is, a linear shape model, shapes that lie between the two clusters have a high probability. However, this region of the input space is not populated by training examples. Instead of learning a Gaussian probability distribution which is not able to adequately capture different clusters, nonlinear learning techniques such as Gaussian Mixture Modeling or Kernel PCA can be used in order to describe the shape distribution more accurately.

The remainder of this section explains the algorithm for vertebra segmentation. In the experiments, a linear Probabilistic ASM is compared against nonlinear versions in order to evaluate whether the use of nonlinear shape modeling techniques has a positive impact on the segmentation accuracy.



(a) Exemplary features of appearance model

(b) Cross-validation of appearance model

Figure 6.12.: (a): Three exemplary 1D features and their encoding as 4-tuple. The feature value is computed by subtracting the intensity sum of the black region from the intensity sum of the white regions. Intensities in dashed regions are ignored. (b): Visualization of the confidence of the local appearance models, estimated using leave-one-out cross validation on the training set. A confidence of 1 means that the appearance model discriminates perfectly between boundary and non-boundary features, whereas a confidence of 0.5 means that the model makes a random guess.

6.3.1. Methodology

The following approach is used to segment a vertebra: The mean shape of the training shapes is manually placed on the target vertebrae in the image using an interactive tool which allows to modify translation, rotation and scale. Table 6.2 gives quantitative measures of the accuracy of this manual initialization. After this initialization, the vertebra is segmented with the Probabilistic ASM, using either a linear or a nonlinear global shape energy. The Probabilistic ASM terminates after a fixed number of 40 iterations. The appearance model that is used in order to detect the boundary is described in the following section.

6.3.1.1. Appearance Model

The local appearance model at a landmark is based on one-dimensional intensity profiles, which are oriented along the landmark's normal. A profile is described by a seven dimensional feature vector $\boldsymbol{f}$ which consists of seven sampled intensity values. An overcomplete set of features is defined for describing such profiles. The set contains 1D Haar-like features as a subset, but also some additional features. Each feature $H = (j, l_1, l_2, l_3)$ is uniquely defined by an offset j and three subprofile lengths $l_1 \geq 1$ and $l_2, l_3 \geq 0$. The feature value of H is defined by

$$h(\boldsymbol{f}, H) = \sum_{i=j}^{\hat{l}_1 - 1} f_i - \sum_{i=\hat{l}_1}^{\hat{l}_2 - 1} f_i + \sum_{i=\hat{l}_2}^{\hat{l}_3 - 1} f_i \tag{6.8}$$

with $\hat{l}_k = j + \sum_{i=1}^{k} l_k$. Three examples of such features, including the encoding as a 4-tuple, are shown in Figure 6.12(a).

In the training phase of the appearance models, a set of training profiles is extracted from the training images for each landmark. Some of the profiles are centered on the organ's boundary, while others are located in the vicinity of the boundary, but are slightly displaced from it. The former profiles are labeled as positive, and the latter as negative examples. These training profiles are used to train a boosted classifier that discriminates between boundary and non-boundary profiles based on the features described above. A strong classifier is learned using the variant of AdaBoost proposed by Viola and Jones [VJ01], which selects only the most discriminative features, and discards the non-discriminative ones. Decision tree stumps are used as weak classifiers.

During segmentation, an intensity profile is sampled for each candidate position of a landmark. The trained classifier is used to compute the probability that a given profile lies on the boundary. Optimal surface detection [HMMW07] (see Section 2.4.3.3) is used in order to select a consistent set of image features.

Figure 6.12(b) visualizes classification rate of the local appearance models at the landmarks, which has been estimated using leave-one-out cross validation on the training data. Especially at the vertebral body, the classification rate is high, that is, the appearance model discriminates here very well between boundary and non-boundary image features. The accuracy is worse at the vertebral processes in the anterior region of the vertebra, because this is where a vertebra adjoins to other bones.

6.3.2. Experiments

The goals of the experiments are twofold: The first goal is two evaluate the performance of the Probabilistic ASM for vertebra segmentation. The second goal is to investigate whether using a nonlinear shape model in this application increases the segmentation accuracy as compared to the commonly used linear shape model.

In order to evaluate the general performance of the linear Probabilistic ASM for vertebra segmentation, it is compared against the standard ASM. Both Probabilistic ASM and standard ASM are executed using the same initialization and the same appearance model. All common parameters are set to identical values. The segmentation accuracy of the Probabilistic ASM depends on the choice of the balancing parameter α that balances image information and prior knowledge. This parameter is manually tuned by trying a large range of different values.

For the comparison of linear and nonlinear shape priors, the Probabilistic ASM is executed with several Kernel PCA-based shape models using a Gaussian radial basis function kernel (Equation 5.5). The actual shape energy depends on the choice of the kernel's hyperparameter σ_{scale}. An initial range of possible values for σ_{scale} has been determined using an interactive tool which plots the fitness landscape of the shape energy for the first two principal components. Based on this initial range, the following ten parameters are used for σ_{scale}: $0.5, 1, 2, \ldots, 9$. For every parameter, a different balancing parameter

α must be chosen. As it is done for the linear shape model, this parameter is tuned manually for each of the ten kernels.

6.3.3. Results

Selected quantitative segmentation results computed during the experiments are given in Table 6.2. It shows the segmentation accuracy—measured with the metrics explained in Section 2.6—of the standard ASM, the linear Probabilistic ASM and the nonlinear Probabilistic ASM with four different kernels. For each Probabilistic ASM, the shown measures are those obtained when using the respective optimal choice for the balancing parameter α. It can be observed that the linear Probabilistic ASM is considerably more accurate than the standard ASM: For example, the ASD is in average 0.23 mm smaller. A qualitative comparison of the two algorithms is given in Figure 6.14. The running time of the linear Probabilistic ASM is comparable to the standard ASM, albeit slightly slower.

For most of the tested Gaussian kernels, the segmentation accuracy of the nonlinear Probabilistic ASM is virtually identical to that of the linear version, as can be seen in Table 6.2 and Figure 6.13. The main difference is that the Hausdorff Distance is in average slightly smaller when using a nonlinear shape prior. On the other hand, the running time of the nonlinear Probabilistic ASM is approximately twice as long as that of the linear counterpart. Figure 6.15 shows Dice coefficient and ASD of the standard ASM as well as for the linear and a nonlinear Probabilistic ASM broken down to the two vertebra types, lumbar and thoracic vertebrae. One can see that all algorithms perform slightly better on lumbar vertebrae, but no ASM variant is particularly accurate or inaccurate on a specific vertebra type.

For narrow kernels ($\sigma_{\text{scale}} = 0.5, 1$), the segmentation accuracy decreases, whereas the running time increases. In these experiments, the library used for optimization reports numerical difficulties when computing the gradient of the shape energy.

Dependent on the used shape prior, a different balancing parameter is optimal. Figure 6.16 shows how the segmentation accuracy varies for different Probabilistic ASM variants when changing α. The graph shows that the narrower the kernel is, the higher α must be chosen in order to obtain an accurate segmentation.

6.3.4. Discussion

The results of the experiments show that the Probabilistic ASM segments vertebrae more accurately than the standard ASM. The improved accuracy can be accounted mainly to the fact that the Probabilistic ASM is more flexible and is therefore able to adapt the model closer to the organ boundary detected by the appearance model. A second reason for the better accuracy is probably the improved specificity of the Probabilistic ASM: Because the algorithm also penalizes deviations from the mean shape, implausible shapes like the one shown in Figure 6.14(c) can be better excluded.

Table 6.2.: Average segmentation accuracy and standard deviation (SD) of the standard ASM, the linear Probabilistic ASM and the Kernel PCA-based Probabilistic ASM with four different Gaussian kernels on the vertebrae data set. For each Probabilistic ASM (ProASM) variant, α was chosen optimally (linear: 0.0415, $\sigma_{\text{scale}} = 1$: 210, $\sigma_{\text{scale}} = 3$: 13, $\sigma_{\text{scale}} = 6$: 1.975, $\sigma_{\text{scale}} = 9$: 0.3). The first row gives the accuracy of the manual initialization, which has been used for each experiment.

	σ_{scale}	Dice	RVD [%]	ASD [mm]	RMS [mm]	HD [mm]	Time [s]
Initial		0.58 SD: 0.09	19.11 SD: 44.11	3.55 SD: 1.0	4.51 SD: 1.2	15.83 SD: 4.0	
standard ASM		0.87 SD: 0.06	-8.78 SD: 7.7	0.90 SD: 0.6	1.50 SD: 1.1	9.42 SD: 5.0	25 SD: 1
ProASM (linear)		0.90 SD: 0.3	-11.35 SD: 6.1	0.67 SD: 0.3	1.25 SD: 0.6	9.21 SD: 3.2	29 SD: 1
ProASM (KPCA)	1	0.88 SD: 0.05	-11.00 SD: 7.5	0.83 SD: 0.4	1.59 SD: 0.7	11.52 SD: 4.6	182 SD: 32
	3	0.90 SD: 0.03	-11.00 SD: 6.1	0.66 SD: 0.3	1.21 SD: 0.6	8.97 SD: 3.73	50 SD: 5
	6	0.90 SD: 0.03	-10.89 SD: 7.3	0.69 SD: 0.4	1.24 SD: 0.7	8.95 SD: 4.0	49 SD: 3
	9	0.90 SD: 0.02	-10.06 SD: 5.0	0.69 SD: 0.2	1.24 SD: 0.3	8.12 SD: 2.6	54 SD: 4

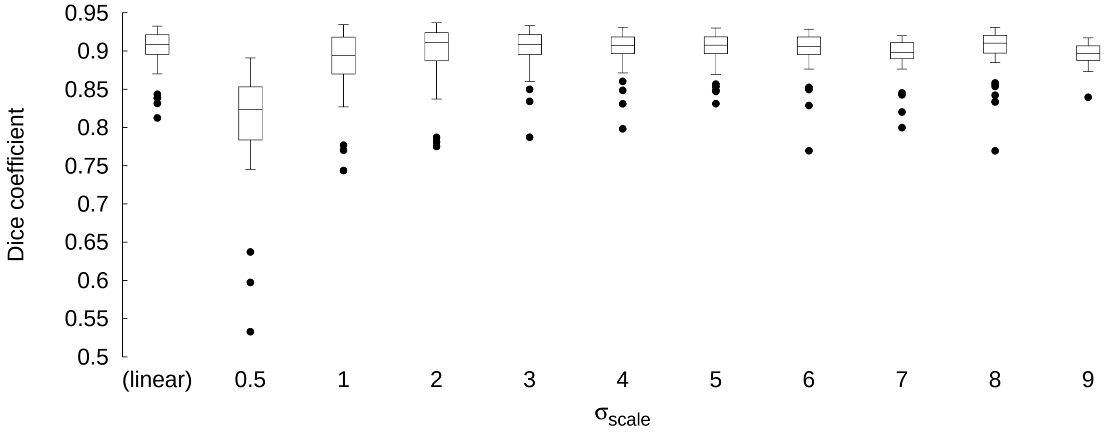
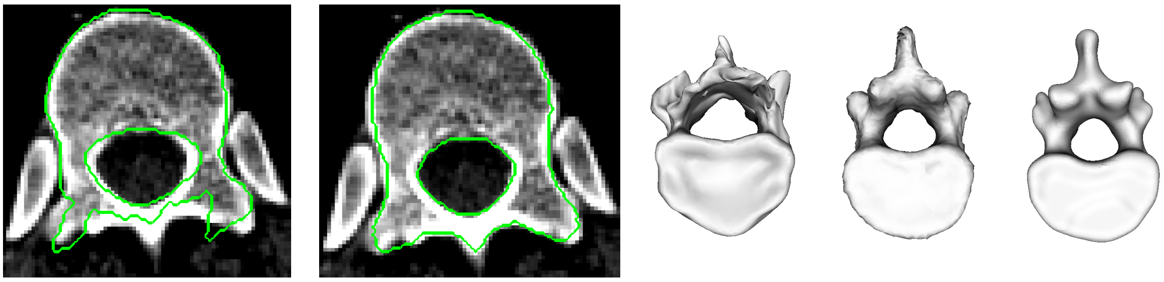


Figure 6.13.: Box plots of Dice coefficient achieved by the linear Probabilistic ASM and the nonlinear Probabilistic ASM with kernels with varying width. The results shown are here are those obtained with optimally chosen balancing parameter α for each shape prior, respectively.



(a) ASM (b) ProASM (c) ASM mesh (d) ProASM mesh (e) Ground truth

Figure 6.14.: Qualitative comparison of the segmentation accuracy of the standard ASM (a) and the linear Probabilistic ASM (b). The segmentation contour is shown in red, and the ground truth in green. The segmentation of the Probabilistic ASM is much more accurate, especially in the posterior region. In this example, the final segmentation mesh computed by the ASM, which is shown in (c), is implausible. The mesh computed by the Probabilistic ASM (d) is much more similar to the ground truth mesh (e).

The segmentation accuracy of the Probabilistic ASM also compares favorably to the results of Klinder et al. [KOE*09], who report an ASD of more than 1.0 mm for thoracic vertebrae (Th10, Th11, Th12) and of more than 0.74 mm for Lumbar vertebrae (L1, L2, L3). It must be noted that Klinder et al. use automatic initialization, while manual initialization is used in the experiments here. Inaccurate automatic initialization potentially affects the robustness of the segmentation. On the other hand, Klinder et al. use a dedicated model for each vertebra type, whereas only one model is used in this work, such that image-guided model selection is not necessary.

Even though the training shapes can be divided into two distinct clusters, the use of nonlinear shape priors does not notably improve the segmentation accuracy. There are several possible explanations for this observation: Firstly, only 23 training shapes have been used for learning the shape model, which may not be sufficient for reliable density estimation. Secondly, the segmentation is driven by image information. That is, it depends also on the image which intermediate shapes are observed during segmentation. The image information itself may exclude that some shapes occur which the shape model would deem as plausible, but do not correspond to a natural shape.

For very narrow Gaussian kernels, the segmentation accuracy decreases considerably. There are two possible explanations for this observation: The first explanation is that a narrow kernel has the risk that the learned distribution overfits the training data. In the worst case, a local energy minimum is created for each training data point. In that case, the shape prior looses flexibility and the risk increases that the segmentation gets stuck in local optima. The second explanation is a more technical one: The results may be affected by the numerical problems reported by the optimizer. Note that a Gaussian kernel numerically evaluates to zero if the distance between two shapes is large. This effect is enforced when the selected scale parameter is small.

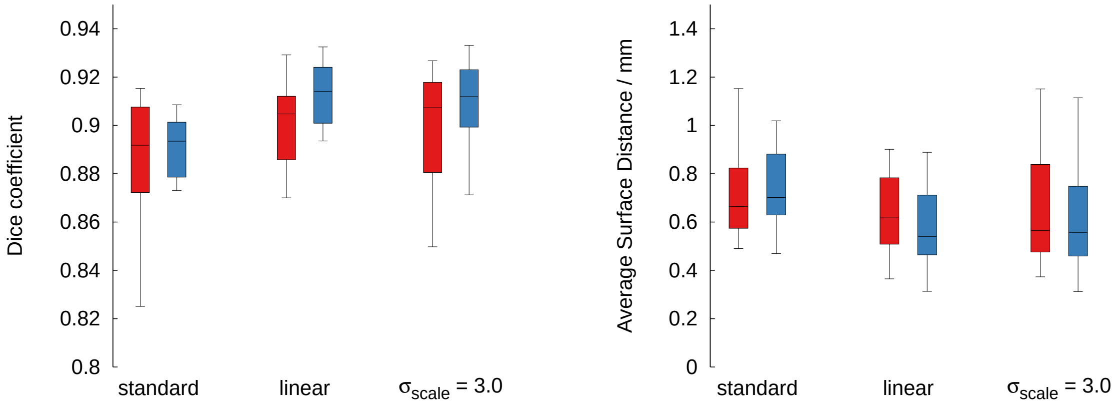


Figure 6.15.: Box plots of Dice coefficient and ASD of the standard ASM, the linear Probabilistic ASM and the nonlinear Probabilistic ASM with $\sigma_{\text{scale}} = 3$. The results are shown for each vertebra type separately: Red boxes show the results for thoracic, and blue boxes the results for lumbar vertebrae.

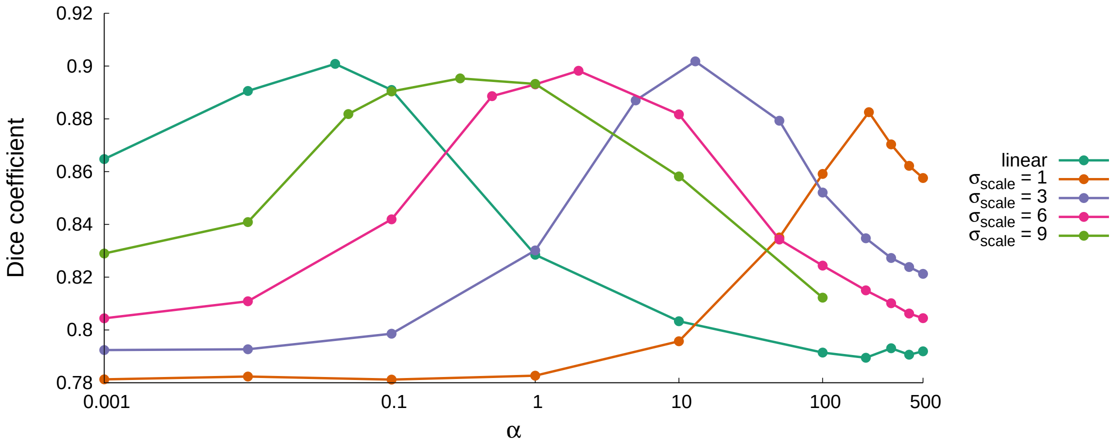


Figure 6.16.: The plot shows how the segmentation accuracy varies for different Probabilistic ASMs when changing the balancing parameter α. The circles are placed at actual measurements.

The results of the experiments made in this thesis are slightly different to those of a preceding study [KBW11], in which the segmentation accuracy of nonlinear shape models is slightly better than that of linear models. However, the preceding study uses fewer test images, and does not employ a local shape model.

In conclusion, the experiments do not show that using nonlinear shape models is advantageous in this application. While the segmentation accuracy achieved with the linear and the best nonlinear model are comparable, the Probabilistic ASM is considerably faster when using a linear model. This is because a shape must be compared to all training examples when computing the nonlinear, Kernel PCA-based shape energy. Moreover, using nonlinear models requires to select an appropriate kernel function which describes the data set well. This means that additional parameters, such as the scale factor σ_{scale} of the Gaussian kernel, must be tuned. However, the experiments do not allow to conclude that nonlinear shape modeling is completely irrelevant for medical image segmentation. The relative performance of nonlinear models may improve if more training examples are available, such that density estimation becomes more reliable. Secondly, the importance of more specific, nonlinear shape priors may be higher in cases where the images are very noisy and the image information is therefore less reliable.

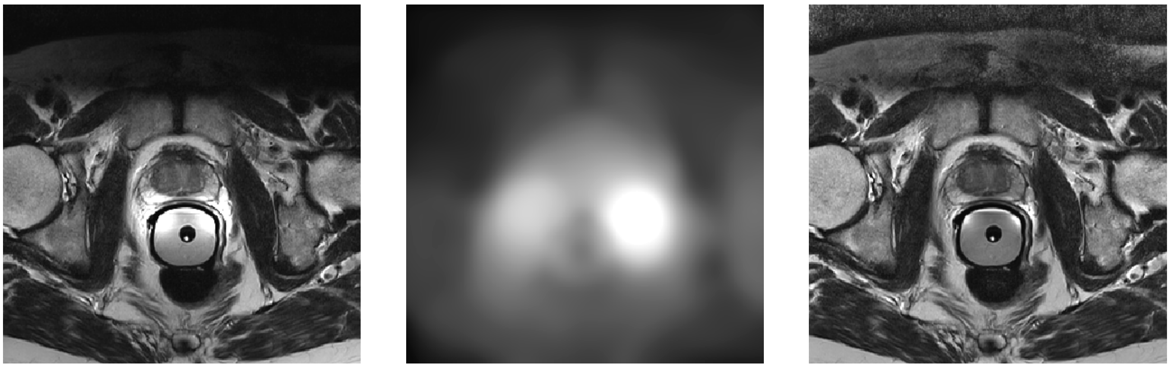
6.4. Prostate segmentation

Prostate segmentation in MRI scans has several important applications, such as cancer detection in computer aided diagnosis [VBG*08, MPL*09], planning of radiation therapy [KvdHL*08] and prostate brachytherapy [MDT08]. Segmentation of the prostate in MRI scans is hampered by various properties of the image data: The prostate has relatively low contrast to surrounding structures and the tissue of the prostate itself appears quite heterogeneous. This heterogeneity is even further increased by cancerous regions—note that cancer is often the reason why the images are acquired in the first place. In contrast to CT imaging, MRI does not provide standardized image intensity such that modeling the prostate's appearance becomes particularly challenging. Due to local differences in the strength of the magnetic field, a spatially smooth intensity variation can often be observed in an MR image, as for example in the image in Figure 6.17(a). This so-called *bias field* causes that the same tissue type is mapped to very different intensity values in the same image. Finally, images acquired with scanners of different vendors or different scanning protocols vary significantly in appearance. For example, some images are acquired using an endorectal coil, which is itself clearly visible in the data, and others not.

The relevance of prostate segmentation in MRI scans is emphasized by the fact that several segmentation challenges have been organized which are particularly dedicated to this topic. The *MICCAI Grand Challenge: Prostate MR Image Segmentation 2012* (PROMISE12)[2] workshop was held in October 2012. The MR data provided for participants of this challenge reflects all challenges of MR data described above. A second segmentation workshop with this topic is held in May 2013, co-located with the IEEE ISBI conference.

---

[2] http://promise12.grand-challenge.org/



(a) Original scan (b) Estimated bias field (c) Corrected image

Figure 6.17.: Prostate MRI scan from the PROMISE12 challenge. (a): The image intensities in the upper part of the original scan are lower than in the rest of the image due to inhomogeneities of the magnetic field. (b): Bias field estimated by the CLIC method [LXAG09]. (c): Corrected image, obtained by dividing the original scan through the bias field.

State-of-the-art approaches for prostate segmentation rely heavily on prior knowledge. Typical approaches use shape modeling [MPL*09, TM12, VGB12, BZR*12] or atlas registration [MDT08, KvdHL*08, ODED12]. In the following, an application of the Probabilistic Active Shape Model to prostate segmentation in MRI scans is described.

6.4.1. Methodology

The segmentation pipeline for prostate MRI scans consists of three steps: In the preprocessing phase, inhomogeneities in the MRI data are corrected and the intensities are normalized into a standardized intensity range. The prostate is then detected on the preprocessed data and segmented using the Probabilistic ASM. The algorithm uses the appearance model based on 1D Haar-like features that has also been used for vertebra segmentation in Section 6.3.1.1. The following paragraphs give further information about preprocessing and prostate detection.

6.4.1.1. Bias field correction and intensity normalization

Image intensities in MRI data are not standardized, that is, the intensity of a certain tissue type varies significantly from image to image. An additional acquisition artifact of MRI is a smoothly varying intensity inhomogeneity within a single scan, the so-called *bias field*, which is caused by inhomogeneous strength of the magnetic field. The bias field is often modeled as a multiplicative component of the image intensities [VPL07]. As both non-standardization and the bias field impede accurate and robust segmentation, image preprocessing and intensity normalization are necessary precursors for robust segmentation.

The bias field is removed using *coherent local intensity clustering* (CLIC) [LXAG09]. CLIC is a segmentation-based approach for bias field correction which jointly estimates the bias field and assigns voxels to one of k different tissue classes which are described by their mean intensities. The algorithm minimizes an energy that depends on the difference between the intensities of adjacent voxels and the mean intensities of the tissue classes to which they have been assigned. Minimization is done using iterative local Fuzzy C-Means clustering which alternates between assigning voxels to intensity classes, updating the mean intensities of the tissue classes and calculating the bias field. The only assumption made about the bias field is that it is smooth. The degree of smoothness is controlled by the size of a Gaussian kernel.

CLIC depends on several parameters: The scale of the truncated Gaussian which defines the size of the local kernel is set to $\sigma = 4$, and $q = 2$ is used as fuzzifier for the Fuzzy-C-Means. These values have been chosen because they have been recommended by Li et al. [LXAG09]. Moreover, $k = 4$ different intensity classes are used, as this choice gave visually the best results. Because the algorithm converges relatively slowly, it is terminated after a fixed number of 20 iterations. Further processing does not have a notable effect on the results. An exemplary application of the CLIC image to a prostate MR scan is illustrated in Figure 6.17, which shows the original scan, the estimated bias field as well as the corrected image.

After bias field computation, the image intensities are normalized and rescaled. A common approach for image normalization is to shift the mean intensity to zero and scaling the intensities such that the standard deviation equals one. This kind of normalization lacks robustness in case of outliers. In the prostate MRI data, such outliers occur often near the endorectal coil, where some voxels have considerably higher intensities than the voxels in the rest of the images. Therefore, mean intensity and standard deviation are estimated using robust statistics, that is, the median image intensity I_{med} is used as an estimator for the mean intensity, and the median absolute deviation MAD_I as an estimator for the standard deviation. The median absolute deviation MAD_I is defined by

$$\text{MAD}_I = \text{median}_{\boldsymbol{v}} \left| I(\boldsymbol{v}) - I_{\text{med}} \right|, \tag{6.9}$$

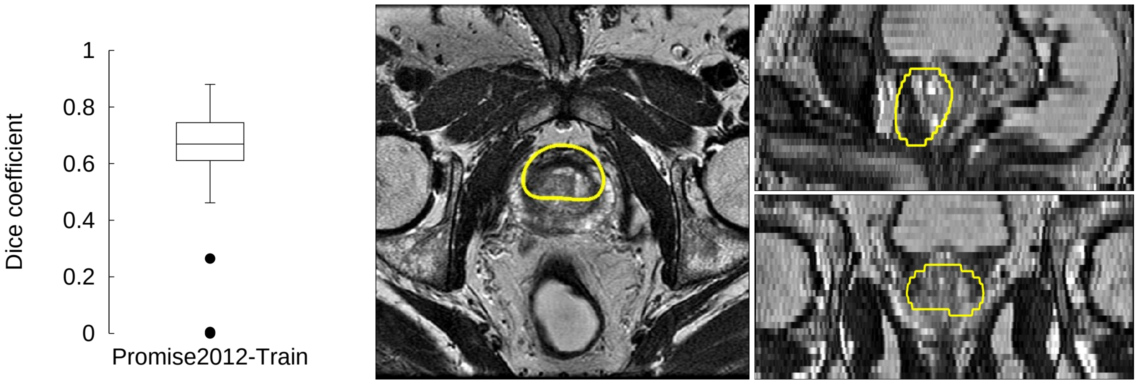
where $I(\boldsymbol{v})$ denotes the image intensity at voxel $\boldsymbol{v}$. For normal distributed data, $1.4826 \cdot \text{MAD}_I$ is approximately equal to the standard deviation. For normalization, the image intensities are scaled to the interval $[0, 1000]$ using the formula

$$I_{\text{norm}}(v) = \frac{(I(v) - I_{\text{med}}) \cdot 167}{1.4826 \cdot \text{MAD}_I} + 500. \tag{6.10}$$

It is assumed here that all intensities lie within three standard deviations from the mean, that is, within the interval $[0, 1000]$. All other voxels outside the three standard deviations are treated as outliers and are clamped accordingly.

6.4.1.2. Prostate detection

Prostate detection is done using the object detection algorithm presented in Section 6.1. The prostate's bounding box contains only little non-prostate tissue, and the prostate's tissue itself is to homogeneous



(a) Detection results (b) Suboptimal initialization

Figure 6.18.: Quantitative and qualitative results of prostate detection. (a): Dice coefficient between detected bounding box and ground truth bounding box on the 50 images of PROMISE2012-TRAIN. (b): Qualitative example with a Dice coefficient of 0.51. In this case, the initialization is sufficient. The final segmentation has a Dice overlap of 0.88 with the expert segmentation (see Figure 6.19).

to find good detectors. The detector is therefore trained by an enlarged ROI that extends the bounding box by 35 percent in x- and y-direction. An expansion in z-direction is not done because most MRI scans have relatively few slices, such that the regions above and below the prostate are often not contained in the scans.

6.4.2. Experiments

The fully automatic algorithm for prostate segmentation is tested on the 80 T2-weighted prostate MR images from the Promise 2012 Grand challenge. The data consists of 50 training cases PROMISE2012-TRAIN and 30 test cases PROMISE2012-TEST. Please see Appendix B for further information on these data sets.

PROMISE2012-TRAIN is used for training and for parameter tuning. For experiments on the training data, the appearance model is trained using leave-one-out learning, and the prostate detector using leave-5-out learning, because training the latter is considerably more time-consuming.

The quality of the prostate detector is evaluated by quantifying the overlap of the ground truth bounding box with the box determined by the detector. Detection results on the test set PROMISE2012-TEST are only performed visually, because the ground truth is not publicly available. Because inaccurate detection results may corrupt the segmentation results, the ground truth bounding box is used in experiments that aim at finding optimal parameters for the Probabilistic Active Shape Model.

After evaluating detection and segmentation separately, the algorithm is evaluated in its entirety, using the parameters that performed best in the previous tests. For a comparison, PROMISE2012-

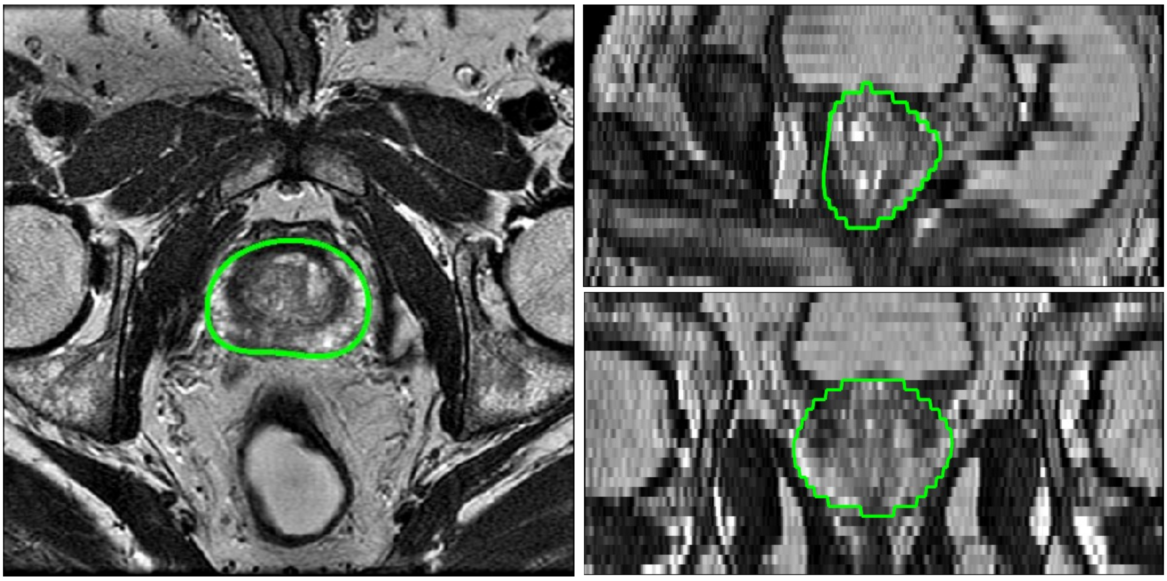


Figure 6.19.: Qualitative segmentation result of the proposed automatic prostate segmentation algorithm based on the Probabilistic ASM. Shown are a transversal slice (left), a sagittal slice (top right) and a coronal slice (bottom right). The Dice overlap with the expert segmentation is 0.88.

TRAIN is also segmented using the standard ASM, using both the automatic detection results and the ground truth bounding box as an initialization.

Results reported on PROMISE2012-TEST have been made with an earlier implementation developed for the PROMISE12 challenge. While the methodology is in principle the same, the improved version used for the experiments on PROMISE2012-TRAIN in this thesis includes bug fixes and differs in some implementational details such that it achieves slightly better results. The submitted segmentations have been evaluated using a scoring system that aggregates several different metrics. Please refer to the challenge's website for further information on score calculation (http://promise12.grand-challenge.org/Details).

6.4.3. Results

The box plot in Figure 6.18(a) gives quantitative detection results on PROMISE2012-TRAIN. The mean Dice coefficient of the detected bounding box with the ground truth bounding box is 0.64. There are three cases in which the detection failed completely (Dice coefficient < 0.3), rendering accurate segmentation impossible. In other cases, the detection result is slightly displaced from the ground truth bounding, as the example shown in Figure 6.18(b).

Figure 6.20 shows the Dice coefficient achieved when segmenting PROMISE2012-TRAIN using different weighting of image and shape energy. The best median Dice coefficient was obtained for

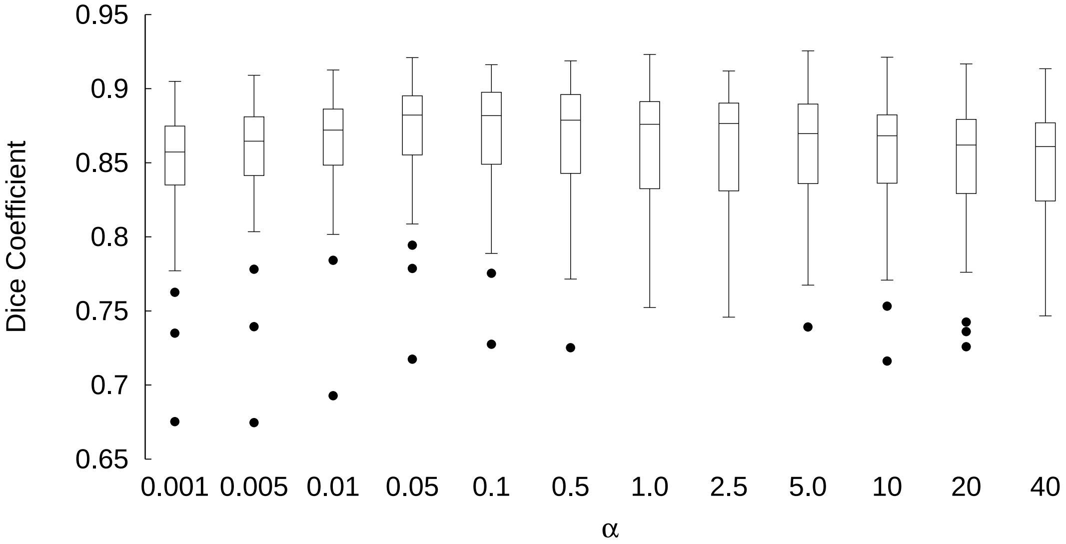


Figure 6.20.: Dice coefficient achieved on PROMISE2012-TRAIN using different weighting of image and shape energy. The higher α, the higher is the influence of the image energy.

$\alpha = 0.05$. The median Dice coefficient becomes worse when choosing a lower or a higher α, and the interquartile range becomes larger. Similar observations can also be made for the surface based evaluation measures ASD, RMS and HD. Therefore, $\alpha = 0.05$ was chosen as balancing parameter for experiments that evaluate the complete segmentation approach.

Table 6.3 shows detailed segmentation results of the standard ASM and the Probabilistic ASM with $\alpha = 0.05$ obtained on PROMISE2012-TRAIN. It includes results for both ground truth initialization and fully automatic prostate detection. Using ground truth initialization, both ASM variants achieve a similar Dice coefficient, but the ASD and the Hausdorff distance of the Probabilistic ASM segmentation are slightly lower. When using automatic detection, there are five cases in which the Dice coefficient achieved by both the Probabilistic ASM and the standard ASM is smaller than 0.6. These cases correspond to those for which the detection is either inaccurate or fails completely. Despite these problematic cases, the median Dice coefficient achieved by the Probabilistic ASM is with 0.87 almost as good as when using ground truth initialization (median Dice: 0.88). In comparison, the performance of the standard ASM degrades considerably when using automatic detection: The median Dice coefficient drops from 0.88 to 0.82. A qualitative result computed by the automatic prostate segmentation algorithm based on the Probabilistic ASM is shown in Figure 6.19.

Segmentation results on the PROMISE2012-TEST of an earlier version of the algorithm have been submitted to the PROMISE12 challenge. The algorithm is ranked on the fifth place out of eleven submissions, and the fourth rank if only automatic algorithms are considered. The average score achieved is 77.6 with a standard deviation of 12.6. The best ranked method [VGB12] achieves an average score

Table 6.3.: Median segmentation accuracy and interquartile range (IQR) of the standard ASM and the linear Probabilistic ASM ($\alpha = 0.05$) on PROMISE2012-TRAIN, using either the bounding boxes of the ground truth or the automatic prostate detector as initialization. Median and IQR are used because of outliers caused by inaccurate automatic initialization. The reported running time is without image preprocessing.

	Dice	RVD [%]	ASD [mm]	RMS [mm]	HD [mm]	Time [s]
Standard ASM						
Ground truth initialization	0.88 IQR: 0.05	4.1 IQR: 19.5	1.4 IQR: 0.6	2.2 IQR: 0.9	9.1 IQR: 3.2	21 IQR: 2
Automatic initialization	0.82 IQR: 0.18	10.3 IQR: 34.0	2.2 IQR: 2.9	3.5 IQR: 3.7	13.9 IQR: 11.6	21 IQR: 2
Probabilistic ASM						
Ground truth initialization	0.88 IQR: 0.04	1.7 IQR: 17.7	1.3 IQR: 0.5	2.0 IQR: 0.7	8.6 IQR: 3.1	25 IQR: 2
Automatic initialization	0.87 IQR:0.09	3.1 IQR: 16.7	1.3 IQR: 1.1	2.1 IQR: 1.4	8.7 IQR: 3.7	25 IQR: 2

of 84.4. The standard deviation of the score of all approaches that are ranked higher is smaller than that of the presented approach.

The running time of the algorithm is between 6-8 minutes per scan using hardware configuration C1 (see Appendix B.4). Most of the time is spent for image preprocessing, that is, for bias field correction. Prostate detection requires less than a second, and shape model adaption takes approximately half a minute.

6.4.4. Discussion

The proposed approach provides accurate means for prostate segmentation in T2-weighted MRI scans. Both Probabilistic ASM and standard ASM achieve a high segmentation accuracy when a good model initialization is given. The Probabilistic ASM has a slightly higher segmentation accuracy, which can be accounted to its increased flexibility. It also shows that the devised appearance model is reliable, because additional flexibility implies a higher risk that the model is being attracted to wrong boundaries.

The robustness of the approach depends to a great extent on the initialization provided by the prostate detector, which is not always reliable. It is likely that the number of training examples (45 for PROMISE2012-TRAIN and 50 for PROMISE2012-TEST) is too small to obtain a robust detector. This lack of robustness is also reflected in the results in the PROMISE12 challenge: The rather large standard deviation of the results are likely to be caused by suboptimal initializations in some scans. The robustness might be increased by exploiting additional knowledge about the scanning protocol: As an example, one could include a position prior that assumes that the prostate is more likely to be located in the center of the scan than in one of its borders. On the other hand, clinical application would require in any case that the segmentation is confirmed by a clinical expert. This means that cases in which the segmentation has failed can be identified, and the segmentation can be restarted with manual initialization. Note that an initialization such as the ground truth bounding box can be easily provided by a human in less than a minute, because it only requires to specify six slices: two transversal, two sagittal, and two coronal slices.

The segmentation results obtained using the fully automatic ASM variants indicate that the Probabilistic ASM is more robust against suboptimal initializations than the standard ASM. Even though both algorithms are not always able to compensate for a bad initialization, the performance of the Probabilistic ASM does not degrade as strongly as that of the standard ASM when using the fully automatic approach. The reason for this is presumably the improved specificity of the Probabilistic ASM: Because it assigns a probability to each shape, unlikely shapes that are far away from the mean are avoided.

The running time of eight minutes per scan is arguably fast enough for an application in clinical practice. It should be noted that the bottleneck of the implementation is not detection or segmentation, but image preprocessing, in particular the CLIC method of Li et al. [LXAG09] for bias field correction. Although some kind of preprocessing is necessary to compensate for image inhomogeneities and to standardize image intensities, it is likely that this method can be replaced by faster methods that achieve equally good results.

7. Conclusion

This thesis presented new methods for shape model-based medical image segmentation, covering topics from model construction by establishing correspondence, over modeling shape distributions to flexible image segmentation. This chapter provides a summary of the contributions of the thesis in Section 7.1, and states open problems and new ideas for future research based on this work in Section 7.2.

7.1. Summary and discussion of contributions

This work made several contributions in two major fields: Shape model construction and shape model-based image segmentation. In the following, the main contributions are summarized and discussed.

7.1.1. Model construction

Construction of shape models by establishing correspondence is indispensable for shape modeling. This thesis presented several new methods for establishing correspondence: Spherical parameterization by Parameter Space Propagation, alignment of spherical shape parameterization, genus 1 parameterization with subsequent correspondence optimization and model construction with nonrigid mesh alignment. The pairwise registration-based techniques have been combined with shape similarity trees in order to improve the correspondence by determining a suitable registration order.

The evaluation of the methods has shown that all proposed techniques are capable of constructing high quality shape models. Which of the method is chosen for model construction depends on the application. For example Parameter Space Propagation is the method of choice for most shapes with spherical topology, while nonrigid mesh alignment stands out especially because of its high versatility and its independence of topology.

The evaluation also includes a comparison with groupwise optimization. The evaluation has shown that by combining good pairwise registration-based methods with shape similarity trees, almost equally good models can be constructed as with groupwise optimization. The main advantage of the registration-based methods is that they are much faster: Whereas groupwise optimization often requires several weeks, registration along shape similarity trees is done within a few hours.

In conclusion, the methods for establishing correspondence devised in this thesis provide means for constructing high quality models of virtually every organ with a distinct shape.

7.1.2. Shape-model based image segmentation

This thesis presented a new algorithm for shape model-based image segmentation, the Probabilistic Active Shape Model. In contrast to previous ASM variants, shapes are modeled by probability distributions. Instead of distinguishing between plausible and implausible shapes, the algorithm seeks for a compromise between image information and prior knowledge about an object's shape. The algorithm allows slight deviations from the learned shape variation in a mathematically well-founded way. This additional flexibility is a key requirement for accurate segmentation. Moreover, the Probabilistic ASM allows to integrate nonlinear shape priors. No other ASM variant has been proposed so far that allows to seamlessly exchange linear and nonlinear models.

The Probabilistic ASM has been evaluated in three application to medical image segmentation: Segmentation of livers and vertebrae in contrast-enhanced CT scans, and segmentation of the prostate in T2-weighted MRI scans. The application on three different structures with largely varying shape complexity and rigidity shows the broad applicability of the approach. Only a single parameter, which is used for balancing shape and image energy, must be determined in order to adapt the proposed approach for constraining shape constraints to a new application.

In all three applications, the Probabilistic ASM achieves consistently better segmentation results than the standard ASM. The main reason for this higher segmentation accuracy is the increased flexibility, because the shapes of segmentations are no longer restricted to linear combinations of the training shapes. The second reason is a higher specificity: The shape energy does not simply classify shapes into plausible and implausible instances, but assigns a probability to them. Thus, shapes that are far away from the training data, but would still be deemed as plausible by the standard ASM, are effectively avoided. Albeit the running time of the Probabilistic ASM is slightly higher than that of the standard ASM due to the algorithmically more complex method for constraining shapes, the shape model adaption still requires less than a minute in all applications.

The experiments with nonlinear shape priors show that some effort is required for selecting reasonable parameters in order to determine a suitable Kernel function. Moreover, the best segmentation results on vertebrae obtained with Kernel PCA-based models are comparable to those obtained with the linear model. The results suggest that the increased effort required when using nonlinear model does not pay off in practice. However, the evaluation results may change in favor to nonlinear models in other applications, or in cases in which more training examples are available.

In conclusion, the Probabilistic ASM provides means for accurate segmentation with flexible and specific shape priors. In contrast to many other flexible shape model-based segmentation algorithms, neither post-processing nor additional modeling assumptions are required in order to obtain accurate segmentation results. The Probabilistic ASM is easily adaptable to large a variety of applications, because both global and local shape variation are completely learned from the training examples.

7.2. Future work

There are several interesting directions for future research based on this thesis. This section describes a selection of open problems, but also sketches some new ideas how these problems can be tackled. The discussion here addresses three research fields: Improvement of correspondence algorithms, model selection and shape model-based multi-organ segmentation.

7.2.1. Improving correspondence algorithms

The segmentation-based evaluation of correspondence algorithms in Chapter 4 has shown that the correspondence quality has indeed a measurable impact on segmentation accuracy. It is therefore certainly worthwhile to further improve correspondence algorithms, because a high segmentation accuracy is crucial in applications such as minimal invasive surgery, in which damage of critical anatomical structures must be avoided at any cost.

The pairwise registration-based approaches devised in this thesis calculate correspondences predominantly based on the proximity of points: All consistent parameterization methods internally use the ICP, which is driven by nearest neighbor relations of points. Similarly, the nonrigid mesh registration algorithm described in Section 3.5 uses proximity as main criterion for registration. The only additional information that is considered is the angle between surface normals of points, such that points with inconsistent normals are never considered as nearest neighbors regardless of their actual distance.

It is likely that the registration-based algorithms can be improved by considering further geometric features, such as curvature or crest lines. For example, for nonrigid vertebra registration, it could be beneficial to first extract crest lines, start the registration with a similarity term that only considers points on the crest lines until they are properly aligned, and use the remaining mesh points exclusively in the last stage of the registration for fine tuning. As Davies et al. [DTT08b] (page 54) observe, a general problem of using geometric features is that their benefit depends strongly on the application. For example, corresponding points do not necessarily have similar curvature in all training instances. This is also the reason why the use of geometric features has not been studied within this thesis. On the other hand, when trying to improve correspondences for a particular application, identifying and exploiting appropriate geometric features is a promising way for improving the model quality.

Another direction for future work is the improvement of groupwise optimization. One minor issue with this approach is that it does not account properly for the representation error, that is, the difference between landmark vectors and original input shapes. In the evaluation in Chapter 4, it has been observed that the representation error increases with groupwise optimization because the reparameterization leads to local area and angle distortion in the sampling patterns (see also Figure 2.7 on page 38). The optimization process causes this distortion because it tries to simplify the model and therefore moves landmarks away from complex geometrical features. There is no term in the objective function that explicitly tries to avoid this oversimplification. This problem can be addressed by adding a distortion term to the objective function as a regularization force that penalizes large area or angle distortion.

One option here is to use Degener's energy [DMK03], which has been successfully used within this thesis to reduces distortions when computing shape parameterizations.

The main disadvantage of 3D groupwise optimization is that it is notoriously hard to optimize. The optimization is very time consuming and converges only slowly. An even more serious issue is that the quality of the final model is strongly dependent on the quality of the initialization. This is evident from the experimental results on groupwise optimization on the spherical domain (Section 4.4.1.1 and Section 4.4.1.2), in which the optimized model initialized with rigidly aligned spherical parameterization never reaches such a good model quality as Parameter Space Propagation without optimization.

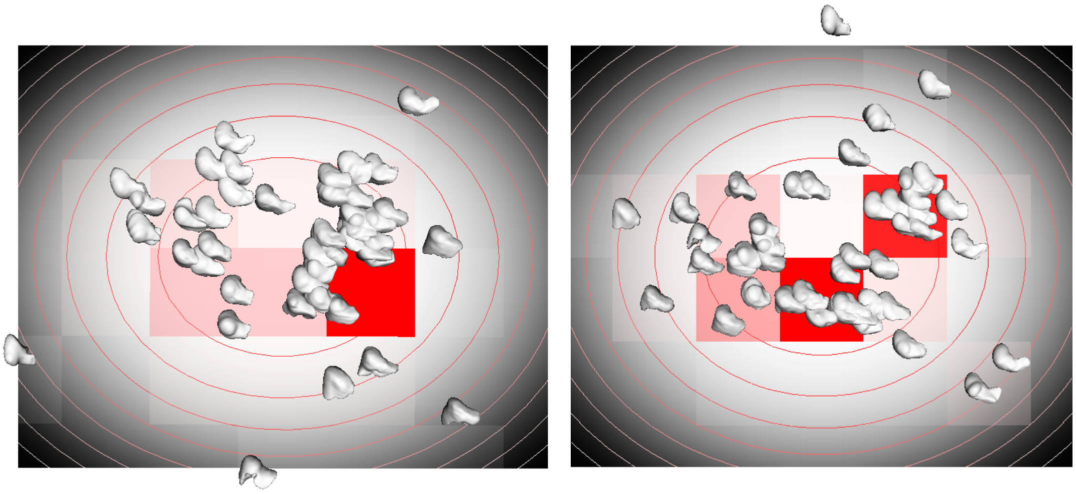
A possible approach for improving groupwise optimization is to increase the problem complexity gradually in order to steer the optimization away from poor local minima and to reduce the convergence time. The idea is to start the optimization not with all shapes at once, but to add them one by one: That is, the optimization considers initially only two shapes, then three, and so on, until all shapes are included. Whenever a shape is added, only its landmarks are subject to optimization, while the landmarks of the remaining shapes are kept fix. Only in the last stage of the optimization, when all shapes are added, a conventional groupwise optimization is used in order to fine tune the correspondences. This optimization technique can be further improved by combining it with shape similarity trees: Using the tree, it can be decided in which order the shapes are added to the optimization. This provides further constraints to steer the complex optimization process.

7.2.2. Model selection

Even though nonlinear shape priors have not improved the segmentation accuracy of vertebra segmentation in Section 6.3, their use may be beneficial in other medical imaging applications. Kernel PCA is an especially flexible method for describing shape distributions, because it is possible to model innumerable many different distributions just by exchanging the kernel function or the kernel's hyperparameters. For example, different vertebra shape distributions have been generated in Section 6.3 by changing the scale factor of the Gaussian kernel function. However, this flexibility has the disadvantage that it becomes necessary to select an appropriate kernel for a given application. This *model selection* problem is nontrivial.

One way to determine optimal parameters is an exhaustive search in the parameter space. This strategy is very time consuming, but still possible as long as the number of free parameters is relatively small. However, it becomes quickly infeasible when the number of parameters increases, for example, if different kernels are used, each with its own set of hyperparameters. But even if there is only a single parameter, such as the scale of a Gaussian kernel function, a more principled approach to model selection is desirable, which does not rely completely on brute force search.

One possibility for model selection is to compute the likelihood of the data, given a particular probability distribution. This means that one chooses the distribution under which observing the training examples is most likely. Cross-validation can be used here in order to improve the reliability. The drawback of this approach is that it is not directly applicable to Kernel PCA: The reason is that Kernel PCA computes a Gaussian distribution in feature space induced by the kernel, not in the input space.



(a) MST - SphAlign

(b) MST - Nonrigid

Figure 7.1.: Visualization of liver shape models constructed using *SphAlign* (a) and *Nonrigid* (b) on a minimum spanning shape similarity tree. The cumulated variance of the first two modes is approximately 41 for the left model, and approximately 29 for the right. The red rectangles in the background are histogram bins. The more saturated a bin is rendered, the more shapes lie within the bin.

Thus, two different kernel distributions are not directly comparable, because the underlying feature space is different.

While the problem of model selection is nontrivial on its own right, it becomes even more challenging in context of shape modeling. The reason for this is that model selection can not be considered independently from the correspondence problem. More particularly, the method used for establishing correspondence may itself influence which shape distribution is considered as optimal for a particular data set. To give an example why this is the case, consider Figure 7.1, which visualizes two sets of landmark vectors computed from the same liver data set. The images show all landmark vectors, projected to the first two principal axes of the respective linear model. In both cases, correspondence has been constructed using a minimum spanning shape similarity tree, but with different correspondence algorithms: In Figure 7.1(a), alignment of spherical parameterizations has been used for establishing correspondence, while nonrigid mesh registration has been used in Figure 7.1(b). One can immediately see from the picture that both projections differ significantly. Both models have a significantly different variance, and even the nearest neighborhood relation between landmark vectors is different. It is likely that a model selection algorithm would propose to use different distributions for both sets of landmark vectors. The question is: Is there a way to decide which of the two distributions—if any—is closer to the 'true' data distribution?

A possible way to do this is to consider how 'stable' a correspondence algorithm is: Given two training sets of the same organ of reasonable size, the optimal distributions estimated by a model selection approach should be approximately the same. Similarly, small modifications of the training set, like removing one of the examples, or small changes of the parameters of the algorithm should not have a large influence. This is presumably not the case for the correspondence algorithms proposed in this thesis: In the experiments made in Chapter 4, changing the template shape influenced the properties of the estimated Gaussian distribution considerably.

7.2.3. Symmetric shape model adaption and multi-organ segmentation

Although the Probabilistic Active Shape Model proposed in this thesis achieves a high segmentation accuracy, the computed segmentations are not perfect. A perfect overlap between automatic segmentation and reference segmentation can hardly be expected: Due to ambiguous image information in medical images, even two manual segmentations of human experts may differ significantly. However, there are cases where the automatic segmentation makes obvious errors that would not have been made by a careful human observer.

For ASM segmentation, a typical reoccurring problem is that parts of the organ are not included in the segmentation. In some cases, these errors can be accounted to large pathologies in the images, for example large liver tumors. A problem here is the local search strategy of Active Shape Models: Because the true organ boundary is far away from the model boundary, it is not visible for the model. Moreover, because of the pathological tumor tissue which appears darker than liver tissue, the model never evolves towards the true boundary. A simple solution to this problem is to enlarge the search radius of the algorithm. However, a larger search radius also implies a larger risk that the model is attracted by boundaries of other organs, which affects the algorithm's robustness negatively.

Another typical error made by ASMs is that small regions of high curvature are not included into the segmentation, even though they have a sufficiently high contrast to their surroundings. This observation can again be accounted to the search strategy. In high curvature regions, the angle between normals of adjacent landmarks of the model mesh can be quite high. This may cause that greater parts of the sought organ boundary lie in a "‘blind angle"’ when the standard Active Shape Model search strategy is used, which samples features on a linear profile oriented along the surface normal. This visibility problem might be alleviated by techniques such as the using the ball-shaped search strategy of Kainmüller et al. [KLH*13], or by using the technique described in Section 3.4 to generate curvature adaptive shape models.

Whichever strategy is used, the fundamental problem remains that current ASM variants employ an *asymmetric* search strategy: Only those parts of the organ boundary are detected that are visible from point of view of the model. A promising direction for future work is to break this limitation by developing a *symmetric* shape model adaption: The basic idea is to depend not only on the set of image features that have been detected by the model, but also detect image features independently from the model, and use these features to pull the model into regions that would be invisible to the model if the conventional, asymmetric strategy was used.

The Probabilistic ASM can be easily adapted to support both kinds of features: Its image energy can be replaced by a symmetric energy, similar to the symmetric similarity term used for nonrigid mesh

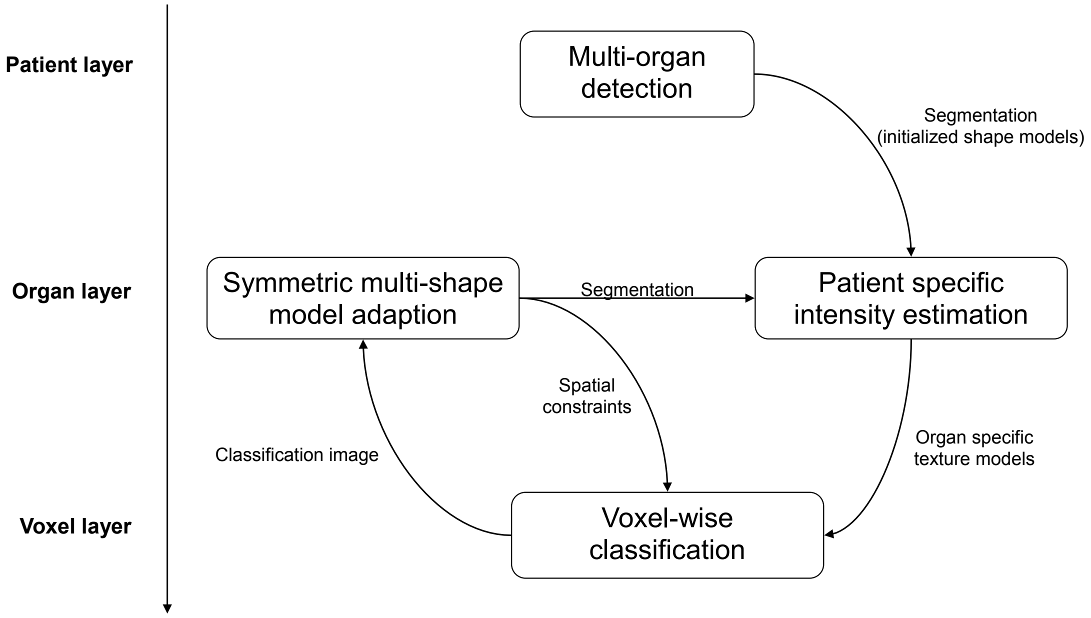


Figure 7.2.: Proposed scheme of a hierarchical segmentation algorithm for automatic multi-organ segmentation by combining object detection, symmetric shape model adaption and voxel classification.

registration proposed in Section 3.5.1. The main challenge is the extraction of features that exert a force on the model: They must belong to the sought organ, because they would misguide the segmentation otherwise and thereby hamper the robustness of the algorithm. Therefore, it is necessary to include additional boundary conditions that prevent the extraction of misguiding image features.

A possible solution to this problem is to simultaneously segment multiple organs. The general idea of segmenting more than one organ at once is not new—in fact, the coupled or articulated models discussed in Section 2.2.5 are often designed for this purpose. The innovative idea proposed here is to enable a symmetric shape model adaption by combining two complementary approaches to multi-organ segmentation: multi-shape model adaption and voxel classification with spatial constraints. Shape model adaption has the advantage that a plausible, topologically consistent segmentation is achieved by the use of global priors. On the other hand, voxel classification methods such as the approaches by Montillo et al. [MSW*11] or Iglesias et al. [IKM*11] have no visibility problem, because every voxel in the image is considered. Based on the classification, regions can be identified that are invisible to the local search strategy of the ASM, but belong with high certainty to a particular organ. Such features are then used to exert a force on the respective shape model. Because the voxel classification is aware of multiple organs and includes spatial constraints describing their relative position, one can expect that this approach is robust, because the risk that the model is drawn towards wrong image features is relatively small.

Figure 7.2 shows a possible structure of an hierarchical multi-organ segmentation approach using symmetric shape model adaption. The proposed scheme uses three hierarchy levels: The topmost level combines state-of-the-art object detection methods and a global location model which describes the relative pose of organs. This level provides a robust initialization of multiple shape models. The actual segmentation is done iteratively by alternating between voxel classification on the bottom layer and symmetric multi-shape model adaption on the middle layer. During voxel classification, a probability image is computed which assigns to each voxel a probability vector, in which each entry encodes the probability that the vector belongs to a particular organ. The shape models are then simultaneously adapted to this probability image. In order to realize a symmetric adaption, voxels having a high membership probability for some organ exert a force on the respective shape model.

Voxel classification is done based on learned appearance models, spatial constraints, and image specific intensity estimation. Spatial constraints are defined by the global location model, which can be instantiated using an intermediate segmentation result. Image specific intensity estimation is used to adapt the learned appearance models to the characteristics of the given image in order to compensate for intensity variations. Such variations are caused, for example, by contrast agents or differences in the image acquisition protocol. Like the spatial constraints, image specific intensity is done on an intermediate segmentation. The hypothesis here is that a rough segmentation is sufficient for estimating the intensity distribution of an organ. Moreover, one can expect that the estimation becomes more and more accurate when the segmentation is refined during the segmentation process. These two hypothesis have already been confirmed in the liver segmentation algorithm presented in Section 6.2.

Such a multi-organ segmentation scheme has several advantages: Firstly, it is to be expected that the segmentation accuracy and robustness of the approach is higher than that of single-organ segmentation algorithms, because the model is aware of several structures, and thus should be able to separate them more accurately. Secondly, the algorithm is more versatile in its applicability, because it is not only useful in applications which focus on a particular organ. Thirdly, due to its versatility, it can be used as a general preprocessing step in order to routinely enrich scans acquired in the clinic with semantic information. This simplifies documentation and later retrieval, for example in studies that investigate the course of a disease or that compare patients with similar diseases.

A. Gradients

This appendix lists the gradients of the energies used in the Probabilistic Active Shape Model (Chapter 5). The gradients are required for energy minimization using the L-BFGS-method [LN89].

A.1. Gradient of linear shape energy

The shape energy is defined as

$$E_{\text{shape}}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{t}\frac{b_i^2}{\lambda_i} + \frac{1}{2\rho}\|\boldsymbol{r}\|^2, \tag{A.1}$$

where $\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b} + \boldsymbol{r}$, $\bar{\boldsymbol{x}}$ is the mean shape, $\boldsymbol{P} \in \mathbb{R}^{3n\times t}$ is the matrix of retained eigenvectors, $\lambda_1 \geq \ldots \geq \lambda_t > 0$ are the corresponding eigenvalues, $\boldsymbol{r}$ is the residual vector and ρ is a regularization parameter. Let $\Lambda^{-1} = \operatorname{diag}(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_t})$. Exploiting that $\|\boldsymbol{r}\|^2 = \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|^2 - \|\boldsymbol{b}\|^2$, the shape energy can be re-written as:

$$\begin{aligned} E_{\text{shape}}(\boldsymbol{x}) &= \frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})^T\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{P}^T(\boldsymbol{x}-\bar{\boldsymbol{x}}) + && \text{(A.2)} \\ &\quad \frac{1}{2\rho}\left[(\boldsymbol{x}-\bar{\boldsymbol{x}})^T(\boldsymbol{x}-\bar{\boldsymbol{x}}) - (\boldsymbol{x}-\bar{\boldsymbol{x}})^T\boldsymbol{P}\boldsymbol{P}^T(\boldsymbol{x}-\bar{\boldsymbol{x}})\right] && \text{(A.3)} \end{aligned}$$

From this, the gradient can be computed using matrix algebra:

$$\nabla E_{\text{shape}}(\boldsymbol{x}) = \boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{P}^T(\boldsymbol{x}-\bar{\boldsymbol{x}}) + \frac{1}{\rho}\left(\boldsymbol{x}-\bar{\boldsymbol{x}}-\boldsymbol{P}\boldsymbol{P}^T(\boldsymbol{x}-\bar{\boldsymbol{x}})\right) \tag{A.4}$$

A.2. Gradient of local shape energy

The local shape energy is defined as

$$E_{\text{local}}(\boldsymbol{x}) = \sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)}\|\boldsymbol{x}^i - \boldsymbol{\mu}^i - \boldsymbol{x}^j\|^2, \tag{A.5}$$

where the $\boldsymbol{x}^i \in \mathbb{R}^d$ are the landmarks and

$$\boldsymbol{\mu}^i = \bar{\boldsymbol{x}}^i - \frac{1}{\mathcal{N}(i)}\sum_{j\in\mathcal{N}(i)}\bar{\boldsymbol{x}}^j. \tag{A.6}$$

Let $\boldsymbol{\mu} = (\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^n)$, $\boldsymbol{\mu} \in \mathbb{R}^{dn}$. Further, the neighborhood relation $\mathcal{N}(i)$ on landmarks is 'translated' to a neighborhood relation on vector entries as follows:

$$\hat{\mathcal{N}}(i) = \{ j \mid j \,\mathrm{div}\, d \in \mathcal{N}(i \,\mathrm{div}\, d) \text{ and } j \equiv i \,(\mathrm{mod}\, d)\}. \tag{A.7}$$

Here, div denotes integer division and mod is the modulo operation. Partial differentiation of $E_{\text{local}}(\boldsymbol{x})$ with respect to x_i yields:

$$\begin{aligned}
\frac{\partial E_{\text{local}}(\boldsymbol{x})}{\partial x_i} &= \sum_{j \in \hat{\mathcal{N}}(i)} 2\,(x_i - \mu_i - x_j) + \qquad (A.8)\\
&\quad \sum_{j \in \hat{\mathcal{N}}(i)} 2\,(x_i - x_j - \mu_j) \qquad (A.9)\\
&= \sum_{j \in \hat{\mathcal{N}}(i)} [4\,(x_i - x_j) + 2\,(\mu_j - \mu_i)] \qquad (A.10)
\end{aligned}$$

A.3. Gradient of image energy

The image energy is

$$E_{\text{image}}(\boldsymbol{x}; \hat{\boldsymbol{x}}; \boldsymbol{w}) = \sum_{i=1}^{N} \|w_i(\boldsymbol{x}^i - \hat{\boldsymbol{x}}^i) + \sum_{j \in \mathcal{N}(i)} w_j(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j)\|^2 \tag{A.11}$$

Partial differentiation with respect to x_i yields:

$$\begin{aligned}
\frac{\partial E_{\text{image}}(\boldsymbol{x}; \hat{\boldsymbol{x}}; \boldsymbol{w})}{\partial x_i} &= 2 \cdot \left[w_i^2(x_i - \hat{x}_i) + \sum_{j \in \hat{\mathcal{N}}(i)} w_i w_j (x_j - \hat{x}_j) \right] + \qquad (A.12)\\
&\quad 2 \cdot \sum_{j \in \hat{\mathcal{N}}(i)} \left[w_i^2(x_i - \hat{x}_i) + w_i w_j (x_j - \hat{x}_j) + \sum_{k \in \hat{\mathcal{N}}(j) - i} w_i w_k (x_k - \hat{x}_k) \right] \qquad (A.13)\\
&= 2(|\hat{\mathcal{N}}(i)| + 1) w_i^2 (x_i - \hat{x}_i) + 4 \cdot \sum_{j \in \hat{\mathcal{N}}(i)} w_i w_j (x_j - \hat{x}_j) + \qquad (A.14)\\
&\quad 2 \cdot \sum_{j \in \hat{\mathcal{N}}(i)} \sum_{k \in \hat{\mathcal{N}}(j) - i} w_i w_k (x_k - \hat{x}_k) \qquad (A.15)
\end{aligned}$$

B. Data sets and hardware

This section gives details about the different data sets used in the experiments conducted in the thesis and the hardware used during evaluation.

B.1. Liver

The liver data consists of 77 contrast enhanced CT scans of thorax and abdomen. The field of view of the scans varies significantly, but the liver is always covered completely. The data has been aggregated from two public and one non-public sources. The non-public part, LOWRES, consists of 27 scans with corresponding expert segmentations. These scans have a relatively coarse resolution, with an intra-slice spacing of 5 mm. The data from the first public source, SLIVER07, are the images from the *MICCAI Grand Challenge* (http://www.sliver07.org/) on liver segmentation. It consists of 20 training images and 10 test images. Expert segmentations are only publicly available for the 20 training images. In order to obtain quantitative results on the test images, the segmentation must be submitted to the challenge's web site. The data from the second public data source, IRCAD, consists of 20 scans, including expert segmentations. This data has been made available under the URL http://www.ircad.fr/softwares/3Dircadb/3Dircadb.php?lng=en by the IRCAD[1] research institute against digestive cancer, Strasbourg, France. An overview over size and spacing of the different subsets of the liver data sets is given in Table B.1.

---

[1] *IRCAD: Institut de recherche contre les cancers de l'appareil digestif*

Table B.1.: Overview over the parameters of the four liver data sets.

Data set	Scans	Voxels per slice	Intra-slice spacing [mm]	Slices	Slice thickness [mm]
LOWRES	27	512×512	0.63 – 0.98	112-148	5
SLIVER07-TRAIN	20	512×512	0.58 – 0.81	64–394	0.7–5
SLIVER07-TEST	10	512×512	0.54 – 0.87	73–502	0.5–3
IRCAD	20	512×512	0.56 – 0.86	74-260	1–4

B.2. Prostate

The prostate MRI data of the PROMISE12 challenge [Pro] is used in the experiments in this work. The data consists of two sets: A set of 50 training images (PROMISE12-TRAIN) and a set of 30 test images (PROMISE12-TEST). The data is *multi-center*, which means that has been acquired in different clinics. Scanners from different vendors and different scanning protocols have been used during image acquisition, such that the data is very heterogeneous. Each image has between 15 to 54 slices with a slice thickness that ranges from 2.2 mm to 4 mm. Each slice contains between 256×256 to 512×512 voxels. The intra-slice spacing ranges from 0.27 mm to 0.75 mm.

For all 80 scans, manual segmentations have been made by human experts with experience with prostate MRI data. These expert segmentations are only publicly available for the 50 training images. In order to evaluate an algorithm on the test images, it is necessary to submit the algorithmic segmentations to the challenge's web site.

B.3. Vertebrae

For vertebra segmentation, 59 thoracic and lumbar vertebrae have been manually segmented in 12 scans of the IRCAD data base (see Appendix B.3).

B.4. Hardware

Due to the large number of tests made in this thesis, the experiments have been conducted on different hardware. The following table lists the different hardware configurations that have been used.

Table B.2.: Hardware configurations used in the experiments.

Name	Processor			RAM	Operating System
	Name	Clock Speed	# Cores		
C1	Intel® Core™ 2 Quad CPU Q9300	2.50GHz	4	8 GB	Windows 7
C2	Intel® Xeon® E5430	2.66GHz	2×4	16 GB	Windows Vista
C3	Intel® Xeon® E5430	2.66GHz	4	8 GB	Ubuntu

C. Publications

This thesis is partially based on the following publications:

1. MATTHIAS KIRSCHNER AND STEFAN WESARG: Area Preserving Parameterisation of Shapes with Spherical Topology. *Informatik 2009: Workshop Medizinische Bildverarbeitung und Mustererkennung*, pp. 1257–1266, Lübeck, 2009.
2. MARIUS ERDT, MATTHIAS KIRSCHNER AND STEFAN WESARG: Simultaneous Segmentation and Correspondence Establishment for Statistical Shape Models. *Proceedings of the 2nd 3D Physiological Human Workshop*, pp. 25–35, Zermatt, 2009.
3. MATTHIAS KIRSCHNER AND STEFAN WESARG: Propagation of shape parameterisation for the construction of a statistical shape model of the left ventricle. *Proceedings of Vision, Modeling, and Visualization*, pp. 313–322, Braunschweig, 2009.
4. MATTHIAS KIRSCHNER AND STEFAN WESARG: Construction of Groupwise Consistent Shape Parameterizations by Propagation. *Proceedings of SPIE Medical Imaging 2010: Image Processing*, San Diego, 2010.
5. MARIUS ERDT, MATTHIAS KIRSCHNER AND STEFAN WESARG: Smart manual landmarking of organs. *Proceedings of SPIE Medical Imaging 2010: Image Processing*, San Diego, 2010.
6. MATTHIAS KIRSCHNER AND STEFAN WESARG: 3D Statistical Shape Model Building Using Consistent Parameterisation. *Bildverarbeitung für die Medizin*, pp. 291–295, Aachen, 2010.
7. MARIUS ERDT, MATTHIAS KIRSCHNER, SEBASTIAN STEGER, STEFAN WESARG: Fast Automatic Liver Segmentation Combining Learned Shape Priors with Observed Shape Deviation. *IEEE International Symposium on Computer-Based Medical Systems*, pp. 249–254, Perth, 2010
8. MATTHIAS KIRSCHNER AND STEFAN WESARG: Active Shape Models Unleashed. *Proceedings of SPIE Medical Imaging 2011: Image Processing*, Lake Buena Vista, 2011.
9. MATTHIAS KIRSCHNER AND STEFAN WESARG: Automatische Initialisierung von Formmodellen mittels modellbasierter Registrierung. *Bildverarbeitung für die Medizin*, pp. 69–73, Lübeck, 2011.
10. MEIKE BECKER, MATTHIAS KIRSCHNER AND STEFAN WESARG: Konsistente Parametrisierung von Flächen vom Geschlecht 1 zur Bildung eines statistischen Formmodells des Wirbels. *Bildverarbeitung für die Medizin*, pp. 74–78, Lübeck, 2011.
11. MARIUS ERDT, MATTHIAS KIRSCHNER, KLAUS DRECHSLER, STEFAN WESARG, MATTHIAS HAMMON, ALEXANDER CAVALLARO: Automatic Pancreas Segmentation in Contrast Enhanced

CT Data Using Learned Spatial Anatomy and Texture Descriptors. *Proceedings of the 8th IEEE International Symposium on Biomedical Imaging*, pp. 2076–2082, Chicago, 2011.

12. Matthias Kirschner, Sebastian T. Gollmer, Stefan Wesarg, Thorsten M. Buzug: Optimal Initialization for 3D Correspondence Optimization: An Evaluation Study. *Information Processing in Medical Imaging*, pp. 308–319, Irsee, 2011.
13. Matthias Kirschner, Meike Becker, Stefan Wesarg: 3D Active Shape Model Segmentation with Nonlinear Shape Priors. *Medical Image Computing and Computer-Assisted Intervention*, pp. 492–499, Toronto, 2011.
14. Meike Becker, Matthias Kirschner, Simon Fuhrmann, Stefan Wesarg: Automatic Construction of Statistical Shape Models for Vertebrae. *Medical Image Computing and Computer-Assisted Intervention*, pp. 500–507, Toronto, 2011.
15. Sebastian Steger, Matthias Kirschner and Stefan Wesarg: Articulated atlas for segmentation of the skeleton from head & neck CT datasets. *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging*, pp. 1256–1259, Barcelona, 2012.
16. Matthias Kirschner and Stefan Wesarg: Regularisierung lokaler Deformation im probabilistischen Active Shape Model. *Bildverarbeitung für die Medizin*, pp. 328–333, Berlin, 2012.
17. Stefan Wesarg, Matthias Kirschner, Meike Becker, Marius Erdt, Konstantinos Kafchitsas and M. Fawad Khan: Dual-energy CT-based Assessment of the Trabecular Bone in Vertebrae. *Methods of Information in Medicine*, Vol.51, 5, pp. 98–405, 2012
18. Matthias Kirschner, Florian Jung and Stefan Wesarg: Automatic Prostate Segmentation in MR Images with a Probabilistic Active Shape Model. *Proceedings of the MICCAI 2012 Grand Challenge on Prostate MR Image Segmentation*, pp. 28–35, Nice, 2012.
19. Florian Jung, Matthias Kirschner and Stefan Wesarg: A generic approach to organ detection using 3D Haar-like features. *Bildverarbeitung für die Medizin*, pp. 320–325, Heidelberg, 2013.
20. Matthias Hammon, Alexander Cavallaro, Marius Erdt, Peter Dankerl, Matthias Kirschner, Klaus Drechsler, Stefan Wesarg, Michael Uder and Rolf Janka.: Model-Based Pancreas Segmentation in Portal Venous Phase Contrast-Enhanced CT Images. *Journal of Digital Imaging*, Vol.26, pp. 1–9, 2013.

D. Supervising Activities

The following list contains the bachelor's, master's and diploma theses supervised by the author.

D.1. Diploma and Master's Theses

1. TOBIAS BAUER: Parallele Ansätze für Active Shape Models. *Diploma thesis*, Technische Universität Darmstadt, 2010.
2. MEIKE BECKER: Parametrisierung und Rekonstruktion einer Fläche vom Geschlecht 1 im Rahmen der Segmentierung mit statistischen Formmodellen. *Master's thesis*, Technische Universität Darmstadt, 2010.
3. FLORIAN JUNG: Automatische Detektion von Organen in CT-Bildern auf Basis des Viola-Jones-Verfahrens. *Master's thesis*, Technische Universität Darmstadt, 2012.
4. DENNIS GEHRKE: Konsistente und verzerrungsminimierende Parametrisierung von Oberflächen sphärischer Topologie. *Diploma thesis*, Technische Universität Darmstadt, 2012.
5. SEBASTIAN FREUTEL: Formmodellbasierte Segmentierung der Leber in kontrastverstärkten CT-Scans. *Master's thesis*, Technische Universität Darmstadt, 2012.

D.2. Bachelor's Theses

1. MICHAEL SCHMITT: Evaluation statistischer Verfahren zur Modellierung von Formvariation. *Bachelor's thesis*, Technische Universität Darmstadt, 2009.
2. PATRICK HÖRMANN: Erstellen statistischer Formmodelle mit Hilfe des EM-ICP-Algorithmus. *Bachelor's thesis*, Technische Universität Darmstadt, 2011.

E. Curriculum Vitae

Personal Data

Name	Matthias Kirschner
Birth date	January 6, 1982
Place of birth	Warburg, Germany
Nationality	German

Education

2008	Master of Computer Science from Universität Paderborn, Germany
2006	Bachelor of Computer Science from Universität Paderborn, Germany
2001	Abitur at Gymnasium Marianum, Warburg

Professional Experience

2008 – 2013	Researcher at the Interactive Graphics Systems Group, Technische Universität Darmstadt, Germany
2005 – 2007	Student software developer at casim GmbH, Kassel
2007	Student research assistant at AG Monien, Universität Paderborn
2006	Student teaching assistant at AG Böttcher, Universität Paderborn

Glossary

List of abbreviations

AAM	Active Appearance Model (Section 2.5.1).
ASD	Average symmetric surface distance (Section 2.6).
ASM	Active Shape Model (Section 2.4).
CDF	Cumulated density function.
CT	Computed tomography.
DIFS	Distance from feature space (Section 2.2.2).
DIFS	Distance in feature space (Section 2.2.2).
EM-ICP	Expectation Maximization Iterative Closest Point.
HD	Hausdorff distance (Section 2.6).
HU	Hounsfield unit.
ICA	Independent Component Analysis.
ICP	Iterative Closest Point.
KPCA	Kernel Principal Component Analysis.
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shannon algorithm.
MDL	Minimum Description Length.
MRI	Magnetic resonance imaging.
PCA	Principal Component Analysis.
RMS	Root mean square symmetric surface distance (Section 2.6).
ROI	Region of interest.
SCDM	Shape Constrained Deformable Model (Section 2.4.4.2).
SSM	Statistical Shape Model.

List of symbols

$\boldsymbol{b}$	Parameter vector of a shape $\boldsymbol{x}$, containing its principal components; $\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b} + \boldsymbol{r}$.
$\boldsymbol{\beta}$	Vector of kernel principal components of a shape.
d	Dimension in which the shapes are embedded.
$k(\boldsymbol{x}, \boldsymbol{y})$	Kernel function; maps two shapes $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{dn}$ to a scalar.
λ_i	The i-th eigenvalue of a shape.
$M(\boldsymbol{x})$	Triangle mesh induced by the landmark vector $\boldsymbol{x}$ and the SSM's connectivity.
$\mathcal{N}(\boldsymbol{i})$	Denotes the set of indices of the neighbors of a mesh point $\boldsymbol{v}_i$.
n	Number of landmarks.
$\boldsymbol{P}$	Eigenvectors of the covariance matrix of the shapes.
$\phi(\boldsymbol{x})$	Function which maps a shape $\boldsymbol{x}$ to a feature space; used in KPCA.
ψ_i	The i-th kernel eigenvalue of a shape.
$\boldsymbol{r}$	Residual vector of the projection of a shape $\boldsymbol{x}$ to the SSM's principal subspace.
s	Number of training shapes.
t	Dimension of the principal subspace of the SSM.
$\mathcal{X}$	Set of training examples; $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s)$.
$\boldsymbol{x}$	Landmark vector $\boldsymbol{x} = (x_1^1, \ldots, x_d^1, \ldots, x_1^n, \ldots, x_d^n) \in \mathbb{R}^{dn}$ with n landmarks of dimension d.
$\boldsymbol{x}^i$	Landmark with dimension d.

Bibliography

[Ale00] ALEXA M.: Merging polyhedral shapes with scattered features. *The Visual Computer 16*, 1 (2000), 26–37. 29, 30

[AN01] ANDRESEN P., NIELSEN M.: Non-rigid registration by geometry-constrained diffusion. *Medical Image Analysis 5* (2001), 81–88. 29, 31, 32

[ANJY06] ABI-NAHED J., JOLLY M.-P., YANG G.-Z.: Robust active shape models: A robust, generic and simple automatic segmentation tool. In *Medical Image Computing and Computer-Assisted Intervention* (2006), Larsen R., Nielsen M., Sporring J., (Eds.), vol. 4191 of *LNCS*, Springer, pp. 1–8. 46, 52

[APH05] ASIRVATHAM A., PRAUN E., HOPPE H.: Consistent spherical parameterization. In *Computer Graphics and Geometric Modeling Workshop* (2005), pp. 265–272. 29, 30

[Arv92] ARVO J.: Fast random rotation matrices. In *Graphics Gems III*, Kirk D., (Ed.). Academic Press Professional, Inc., 1992, pp. 117–120. 67

[ATZ99] AUGUST J., TANNENBAUM A., ZUCKER S.: On the evolution of the skeleton. In *IEEE International Conference on Computer Vision* (1999), vol. 1, pp. 315–322. 11

[Bal81] BALLARD D.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition 13*, 2 (1981), 111–122. 43

[BCP*08] BOISVERT J., CHERIET F., PENNEC X., LABELLE H., AYACHE N.: Geometric variability of the scoliotic spine using statistics on articulated shape models. *IEEE Transactions on Medical Imaging 27*, 4 (2008), 557–568. 25

[BCT*08] BABALOLA K. O., COOTES T. F., TWINING C. J., PETROVIC V., TAYLOR C.: 3D brain segmentation using active appearance models and local regressors. In *Medical Image Computing and Computer-Assisted Intervention* (2008), Metaxas D. N., Axel L., Fichtinger G., Székely G., (Eds.), vol. 5241 of *LNCS*, Springer, pp. 401–408. 55

[Bec10] BECKER M.: *Parametrisierung und Rekonstruktion einer Fläche vom Geschlecht 1 im Rahmen der Segmentierung mit statistischen Formmodellen*. Master's thesis, Technische Universität Darmstadt, 2010. 78

[BF01] BUSS S., FILLMORE J. P.: Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics 20*, 2 (2001), 95–126. 74

[BGK91] BRECHBÜHLER C., GERIG G., KÜBLER O.: Towards representation of 3D shape: Global surface parametrization. In *Visual Form: Analysis and Recognition* (1991), Plenum Press, pp. 79–88. 64

[BGK95] BRECHBÜHLER C., G. G., KÜBLER O.: Parametrization of closed surfaces for 3-D shape description. *Computer Vision and Image Understanding 61*, 2 (1995), 154–170. 37, 64, 65

[BHP06] BADE R., HAASE J., PREIM B.: Comparison of fundamental mesh smoothing algorithms for medical surface models. In *Simulation und Visualisierung* (2006), pp. 289–304. 88

[Bis06] BISHOP C.: *Pattern recognition and machine learning*. Springer, 2006. 15, 16

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 9 (2004), 1124–1137. 49

[BKFW11] BECKER M., KIRSCHNER M., FUHRMANN S., WESARG S.: Automatic construction of statistical shape models for vertebrae. In *Medical Image Computing and Computer-Assisted Intervention* (2011), Fichtinger G., Martel A., Peters T., (Eds.), vol. 6892 of *LNCS*, Springer, pp. 500–507. 78, 99

[BKS*11] BINDERNAGEL M., KAINMUELLER D., SEIM H., LAMECKER H., ZACHOW S., HEGE H.-C.: An articulated statistical shape model of the human knee. In *Bildverarbeitung für die Medizin* (2011), Handels H., Ehrhardt J., Deserno T. M., Meinzer H.-P., Tolxdorff T., (Eds.), pp. 59–63. 24

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (1992), 239–256. 30, 63, 70, 79, 88, 126, 135

[BMS97] BOWDEN R., MITCHELL T., SAHARDI M.: Cluster based non-linear principal component analysis. In *Electronics Letters* (1997), pp. 1858–1859. 21

[BMVS02] BEHIELS G., MAES F., VANDERMEULEN D., SUETENS P.: Evaluation of image features and search strategies for segmentation of bone structures in radiographs using active shape models. *Medical Image Analysis 6*, 1 (2002), 47–62. 49

[BS00] BREJL M., SONKA M.: Object localization and border detection criteria design in edge-based image segmentation: Automated learning from examples. *IEEE Transactions on Medical Imaging 19*, 10 (2000), 973–985. 43

[BT00] BRETT A., TAYLOR C.: A method of automated landmark generation for automated 3D PDM construction. *Image and Vision Computing 18*, 9 (2000), 739–748. 29, 31, 33

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99* (1999), pp. 187–194. 59

[BV01] BRESSAN M., VITRIÀ J.: Independent modes of variation in point distribution models. In *International Workshop on Visual Form* (2001), Arcelli C., Cordella L. P., di Baja G. S., (Eds.), vol. 2059 of *LNCS*, Springer, pp. 123–134. 20

[BV03] BLANZ V., VETTER T.: Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 9 (2003), 1063–1074. 59

[BZR*12] BIRKBECK N., ZHANG J., REQUARDT M., KIEFER B., GALL P., ZHOU S. K.: Region-specific hierarchical segmentation of MR prostate using discriminative learning. In *PROMISE12 - MICCAI 2012 Grand Challenge on Prostate MR Image Segmentation* (2012). 153

[CET01] COOTES T., EDWARDS G., TAYLOR C.: Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 6 (2001), 681–685. 54

[CFK*11] COMBÈS B., FOURNIER M., KENNEDY D., BRAGA J., ROBERTS N., PRIMA S.: EM-ICP strategies for joint mean shape and correspondences estimation: Applications to statistical analysis of shape and of asymmetry. In *IEEE International Symposium on Biomedical Imaging* (2011), pp. 1257–1263. 29, 31

[CFS*07] CATES J., FLETCHER P. T., STYNER M., SHENTON M., WHITAKER R.: Shape modeling and analysis with entropy-based particle systems. In *Information Processing in Medical Imaging* (2007), Karssemeijer N., Lelieveldt B. P. F., (Eds.), vol. 4584 of *LNCS*, Springer, pp. 333–345. 35

[CKS03] CREMERS D., KOHLBERGER T., SCHNÖRR C.: Shape statistics in kernel space for variational image segmentation. *Pattern Recognition 36*, 9 (2003), 1929–1943. 23, 54, 56, 57, 117, 118, 122, 125

[CLRS01] CORMEN T. H., LEISERSON C. E., RIVEST R. L., STEIN C.: *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001. 64, 74, 91

[CLS*07] CUI X., LIU Y., SHAN S., CHEN X., GAO W.: 3D haar-like features for pedestrian detection. In *IEEE International Conference on Multimedia and Expo* (2007), pp. 1263–1266. 128

[Com94] COMON P.: Independent component analysis, a new concept? *Signal Processing 36*, 3 (1994), 287–314. 20

[CP10] COMBÈS B., PRIMA S.: An efficient EM-ICP algorithm for symmetric consistent non-linear registration of point sets. In *Medical Image Computing and Computer-Assisted Intervention* (2010), Jiang T., Navab N., Pluim J. P. W., Viergever M. A., (Eds.), vol. 6362 of *LNCS*, pp. 594–601. 31, 85

[CRD07] CREMERS D., ROUSSON M., DERICHE R.: A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision 72*, 2 (2007), 195–215. 10

[CS09] CHEN J.-H., SHAPIRO L.: PCA vs. tensor-based dimension reduction methods: An empirical comparison on active shape models of organs. In *IEEE Conference on Engineering in Medicine and Biology Society* (2009), pp. 5838–5841. 20

[CT94] Cootes T., Taylor C.: Using grey-level models to improve active shape model search. In *IEEE International Conference on Pattern Recognition* (1994), vol. 1, pp. 63–67. 47

[CT95] Cootes T., Taylor C.: Combining point distribution models with shape models based on finite element analysis. *Image and Vision Computing 13*, 5 (1995), 403–409. 19, 60

[CT97] Cootes T. F., Taylor C. J.: A mixture model for representing shape variation. In *Image and Vision Computing* (1997), BMVA Press, pp. 110–119. 22, 53, 124

[CT01] Cootes T., Taylor C.: Constrained active appearance models. In *International Conference on Computer Vision* (2001), vol. 1, pp. 748–754. 55

[CT04] Cootes T., Taylor C.: Statistical models of appearance for computer vision, 2004. 14, 18, 42, 55

[CTCG95] Cootes T. F., Taylor C. J., Cooper D. H., Graham J.: Active shape models – their training and application. *Computer Vision and Image Understanding 61*, 1 (1995), 38–59. 12, 23, 25, 26, 41, 46, 48, 50, 113, 122

[CTL94] Cootes T., Taylor C., Lanitis A.: Multi-resolution search with active shape models. In *International Conference on Pattern Recognition* (1994), vol. 1, pp. 610–612. 46

[CTWS02] Cremers D., Tischhäuser F., Weickert J., Schnörr C.: Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision 50*, 3 (2002), 295–313. 16, 19, 23, 54, 56, 57, 115, 122, 125

[CVC12] Cerrolaza J., Villanueva A., Cabeza R.: Hierarchical statistical shape models of multiobject anatomical structures: Application to brain MRI. *IEEE Transactions on Medical Imaging 31*, 3 (2012), 713–724. 24

[CVS*12] Cerrolaza J., Villanueva A., Sukno F., Butakoff C., Frangi A., Cabeza R.: Full multiresolution active shape models. *Journal of Mathematical Imaging and Vision 44*, 3 (2012), 463–479. 46

[CZS10] Chen J.-H., Zheng K. C., Shapiro L G.: 3D point correspondence by minimum description length in feature space. In *European Conference on Computer Vision* (2010), Daniilidis K., Maragos P., Paragios N., (Eds.), vol. 6313 of *LNCS*, pp. 621–634. 35

[Dav02] Davies R. H.: *Learning shape: Optimal models for analysing natural variability*. PhD thesis, University of Manchester, 2002. 39, 40

[dBCvKO08] de Berg M., Cheong O., van Kreveld M., Overmars M.: *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer, 2008. 63

[DFP08] DAM E. B., FLETCHER P. T., PIZER S. M.: Automatic shape model building based on principal geodesic analysis bootstrapping. *Medical Image Analysis 12*, 2 (2008), 136–151. 32

[DHKL01] DYN N., HORMANN K., KIM S.-J., LEVIN D.: Optimizing 3D triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces*, Lyche T., Schumaker L. L., (Eds.). Vanderbilt University, 2001, pp. 135–146. 85

[Dic45] DICE L.: Measures of the amount of ecologic association between species. *Ecology 26*, 3 (1945), 297–302. 57

[DLvB*10] DWORZAK J., LAMECKER H., VON BERG J., KLINDER T., LORENZ C., KAINMÜLLER D., SEIM H., HEGE H.-C., ZACHOW S.: 3D reconstruction of the human rib cage from 2D projection images using a statistical shape model. *International Journal of Computer-Assisted Radiology and Surgery 5*, 2 (2010), 111–124. 59

[DMK03] DEGENER P., MESETH J., KLEIN R.: An adaptable surface parametrization method. In *International Meshing Roundtable* (2003). 65, 67, 68, 74, 79, 80, 164

[DRT08] DAMBREVILLE S., RATHI Y., TANNENBAUM A.: A framework for image segmentation using shape models and kernel space shape priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 8 (2008), 1385–1399. 10

[DSSW10] DALAL P., SHI F., SHEN D., WANG S.: Multiple cortical surface correspondence using pairwise shape similarity. In *Medical Image Computing and Computer-Assisted Intervention* (2010), Jiang T., Navab N., Pluim J. P. W., Viergever M. A., (Eds.), vol. 6363 of *LNCS*, pp. 349–356. 33, 90, 91, 95, 101

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (2005), vol. 1, pp. 886–893. 44

[DTC*02a] DAVIES R. H., TWINING C., COOTES T. F., WATERTON J., TAYLOR C. J.: 3D statistical shape models using direct optimisation of description length. In *European Conference on Computer Vision*, Heyden A., Sparr G., Nielsen M., Johansen P., (Eds.), vol. 2352 of *LNCS*. Springer, 2002, pp. 1–17. 35, 37

[DTC*02b] DAVIES R. H., TWINING C. J., COOTES T. F., WATERTON J. C., TAYLOR C. J.: A minimum description length approach to statistical shape modeling. *IEEE Transactions on Medical Imaging 21*, 5 (2002), 525–537. 34, 38

[DTCT10] DAVIES R., TWINING C., COOTES T., TAYLOR C.: Building 3-D statistical shape models by direct optimization. *IEEE Transactions on Medical Imaging 29*, 4 (2010), 961–981. 34, 36, 38, 74

[DTS03] DAVATZIKOS C., TAO X., SHEN D.: Hierarchical active shape models, using the wavelet transform. *IEEE Transactions on Medical Imaging 22*, 3 (2003), 414–423. 19, 24

[DTT06] Davies R. H., Twining C. J., Taylor C. J.: Consistent spherical parameterisation for statistical shape modelling. In *IEEE Symposium on Biomedical Imaging* (2006), pp. 1388–1391. 29, 31, 37, 41, 63, 76, 99

[DTT08a] Davies R. H., Twining C. J., Taylor C. J.: Groupwise surface correspondence by optimization: Representation and regularization. *Medical Image Analysis 12*, 6 (2008), 787–796. 37

[DTT08b] Davies R. H., Twining C. J., Taylor C. J.: *Statistical Models of Shape - Optimisation and Evaluation*. Springer, 2008. 26, 27, 34, 36, 37, 74, 76, 163

[EK07] Ericsson A., Karlsson J.: Measures for benchmarking of automatic correspondence algorithms. *Journal of Mathematical Imaging and Vision 28* (2007), 225–241. 40

[EKSW10] Erdt M., Kirschner M., Steger S., Wesarg S.: Fast automatic liver segmentation combining learned shape priors with observed shape deviation. In *IEEE International Symposium on Computer-Based Medical Systems* (2010), pp. 249–254. 4, 46, 53, 60, 132, 138, 140, 142, 143

[EPS*08] Ecabert O., Peters J., Schramm H., Lorenz C., von Berg J., Walker M. J., Vembar M., Olszewski M., Subramanyan K., Lavi G., Weese J.: Automatic model-based segmentation of the heart in CT images. *IEEE Transactions on Medical Imaging 27*, 9 (2008), 1189–1201. 43, 45, 52

[Erd12] Erdt M.: *Non-Uniform Deformable Volumetric Objects for Medical Organ Segmentation and Registration*. PhD thesis, Technische Universität Darmstadt, 2012. 44

[ESS12] Erdt M., Steger S., Sakas G.: Regmentation: A new view of image segmentation and registration. *Journal of Radiation Oncology Informatics 4* (2012), 1–23. 2

[EW05] Erickson J., Whittlesey K.: Greedy optimal homotopy and homology generators. In *ACM-SIAM Symposium on Discrete Algorithms* (2005), SIAM, pp. 1038–1046. 79, 80

[FAKG10] Fuhrmann S., Ackermann J., Kalbe T., Goesele M.: Direct resampling for isotropic surface remeshing. In *Workshop on Vision, Modeling and Visualization* (2010). 83, 89

[FBM*05] Fripp J., Bourgeat P., Mewes A., Warfield S., Crozier S., Ourselin S.: 3D statistical shape models to embed spatial relationship information. In *Computer Vision for Biomedical Image Applications*, Liu Y., Jiang T., Zhang C., (Eds.), vol. 3765 of *LNCS*. Springer, 2005, pp. 51–60. 29, 31, 37, 70

[FL99] Fleute M., Lavallée S.: Nonrigid 3-D/2-D registration of images using statistical models. In *Medical Image Computing and Computer-Assisted Intervention* (1999), Taylor C., Colchester A. C. F., (Eds.), vol. 1679 of *LNCS*, pp. 138–147. 59

[FLJ99] FLEUTE M., LAVALLÉE S., JULLIARD R.: Incorporating a statistically based shape model into a system for computer-assisted anterior cruciate ligament surgery. *Medical Image Analysis 3*, 3 (1999), 209–222. 59

[FOP*07] FERRARINI L., OLOFSEN H., PALM W. M., VAN BUCHEM M. A., REIBER J. H. C., ADMIRAAL-BEHLOUL F.: GAMEs: Growing and adaptive meshes for fully automatic shape modeling and analysis. *Medical Image Analysis 11*, 3 (2007), 302–314. 59

[Fre12] FREUTEL S.: *Formmodellbasierte Segmentierung der Leber in kontrastverstärkten CT-Scans*. Master's thesis, Technische Universität Darmstadt, 2012. 137

[FRSN01] FRANGI A. F., RUECKERT D., SCHNABEL J., NIESSEN W. J.: Automatic 3D ASM construction via atlas-based landmarking and volumetric elastic registration. In *Information Processing in Medical Imaging* (2001), Insana M. F., Leahy R. M., (Eds.), vol. 2082 of *LNCS*, Springer, pp. 78–91. 29, 31

[FRSN02] FRANGI A. F., RUECKERT D., SCHNABEL J., NIESSEN W.: Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling. *IEEE Transactions on Medical Imaging 21*, 9 (2002), 1151–1166. 29, 30, 31, 32

[FRU*05] FRITZ D., RINCK D., UNTERHINNINGHOFEN R., DILLMANN R., SCHEUERING M.: Automatic segmentation of the left ventricle and computation of diagnostic parameters using regiongrowing and a statistical model. In *SPIE Medical Imaging 2005: Image Processing* (2005), Fitzpatrick J. M., Reinhardt J. M., (Eds.), vol. 5747, pp. 1844–1854. 47

[GB10] GOLLMER S. T., BUZUG T. M.: A method for quantitative evaluation of statistical shape models using morphometry. In *IEEE International Symposium on Biomedical Imaging* (2010), pp. 448–451. 40

[GB12] GOLLMER S. T., BUZUG T. M.: Fully automatic shape constrained mandible segmentation from cone-beam CT data. In *IEEE International Symposium on Biomedical Imaging* (2012), pp. 1272–1275. 41, 52

[GOM*10] GHOSE S., OLIVER A., MARTÍ R., LLADÓ X., FREIXENET J., VILANOVA J., MERIAUDEAU F.: Prostate segmentation with texture enhanced active appearance model. In *International Conference on Signal-Image Technology and Internet-Based Systems* (2010). 55

[Goo91] GOODALL C.: Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B 53*, 2 (1991), 285–339. 14

[Gow75] GOWER J.: Generalized procrustes analysis. *Psychometrika 40* (1975), 33–51. 14

[GP06] GRANGER S., PENNEC X.: Multi-scale EM-ICP: A fast and robust approach for surface registration. In *European Conference on Computer Vision* (2006), Heyden A., Sparr G., Nielsen M., Johansen P., (Eds.), vol. 2353 of *LNCS*, Springer, pp. 69–73. 31, 72, 109

[GW08] GONZALEZ R., WOODS R. E.: *Digital Image Processing*. Prentice Hall, 2008. 66

[HBS*08] HEIMANN T., BAUMHAUER M., SIMPFENDÖRFER T., MEINZER H.-P., WOLF I.: Prostate segmentation from 3D transrectal ultrasound using statistical shape models and various appearance models. In *SPIE Medical Imaging 2008: Image Processing* (2008), Reinhardt J., Pluim J. P. W., (Eds.), vol. 6914. 43

[HEP*09] HUFNAGEL H., EHRHARDT J., PENNEC X., AYACHE N., HANDELS H.: Computation of a probabilistic statistical shape model in a maximum-a-posteriori framework. *Methods of Information in Medicine 48*, 4 (2009), 314–319. 29, 31, 32

[HH96] HEAP A., HOGG D.: Extending the point distribution model using polar coordinates. *Image and Vision Computing 14*, 8 (1996), 589–599. 24

[HH97] HEAP A., HOGG D.: Improving specificity in PDMs using a hierarchical approach. In *British Machine Vision Conference* (1997), Clark A., (Ed.). 21

[HLS07] HORMANN K., LÉVY B., SHEFFER A.: Mesh parameterization: theory and practice. In *ACM SIGGRAPH 2007 courses* (2007), ACM. 36

[HM09] HEIMANN T., MEINZER H.-P.: Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis 13*, 4 (2009), 543–563. 26, 28, 46, 56

[HMMW07] HEIMANN T., MÜNZING S., MEINZER H.-P., WOLF I.: A shape-guided deformable model with evolutionary algorithm initialization for 3D soft tissue segmentation. In *Information Processing in Medical Imaging* (2007), Karssemeijer N., Lelieveldt B. P. F., (Eds.), vol. 4584 of *LNCS*, pp. 1–12. 43, 48, 49, 52, 132, 136, 143, 147

[Hop96] HOPPE H.: Progressive meshes. In *SIGGRAPH'96* (1996), pp. 99–108. 65

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A 4*, 4 (1987), 629–642. 50, 70, 71, 115, 135

[HTB00] HILL A., TAYLOR C., BRETT A. D.: A framework for automatic landmark identification using a new method of nonrid correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 3 (2000), 241–251. 32

[HULR07] HANSEGÅRD J., URHEIM S., LUNDE K., RABBEN S.: Constrained active appearance models for segmentation of triplane echocardiograms. *IEEE Transactions on Medical Imaging 26*, 10 (2007), 1391–1400. 55

[HvGS*09] HEIMANN T., VAN GINNEKEN B., STYNER M., ARZHAEVA Y., AURICH V., BAUER C., BECK A., BECKER C., BEICHEL R., BEKES G., BELLO F., BINNIG G., BISCHOF H., BORNIK A., CASHMAN M. M., CHI Y., CORDOVA A., DAWANT M., FIDRICH M., FURST D., FURUKAWA D., GRENACHER L., HORNEGGER J., KAINMULLER D., KITNEY I., KOBATAKE H., LAMECKER H., LANGE T., LEE J., LENNON B., LI R., LI S., MEINZER H.-P., NEMETH G., RAICU S., RAU A., VAN RIKXOORT M., ROUSSON M., RUSKO L., SADDI A., SCHMIDT G., SEGHERS D., SHIMIZU A., SLAGMOLEN P., SORANTIN E., SOZA G., SUSOMBOON R., WAITE M., WIMMER A., WOLF I.: Comparison and evaluation of methods for liver segmentation from CT datasets. *IEEE Transactions on Medical Imaging 28* (2009), 1251–1265. 4, 12, 57, 132, 142

[HWM06] HEIMANN T., WOLF I., MEINZER H.-P.: Active shape models for a fully automated 3D segmentation of the liver – an evaluation on clinical data. In *Medical Image Computing and Computer-Assisted Intervention* (2006), Larsen R., Nielsen M., Sporring J., (Eds.), vol. 4191 of *LNCS*, Springer, pp. 41–48. 47

[HWM07] HEIMANN T., WOLF I., MEINZER H.-P.: Automatic generation of 3D statistical shape models with optimal landmark distribution. *Methods of Information in Medicine 46*, 3 (2007), 275–281. 37, 38, 40, 83

[HWWM05] HEIMANN T., WOLF I., WILLIAMS T., MEINZER H.-P.: 3D active shape models using gradient descent optimization of description length. In *Information Processing in Medical Imaging* (2005), Christensen G. E., Sonka M., (Eds.), vol. 3565 of *LNCS*, Springer, pp. 566–577. 35, 37

[HYVD10] HAMM J., YE D. H., VERMA R., DAVATZIKOS C.: A framework for geodesic registration on anatomical manifolds. *Medical Image Analysis 14*, 5 (2010), 633–642. 91

[IKM*11] IGLESIAS J. E., KONUKOGLU E., MONTILLO A., TU Z., CRIMINISI A.: Combining generative and discriminative models for semantic segmentation of CT scans via active learning. In *Information Processing in Medical Imaging* (2011), Székely G., Hahn H., (Eds.), vol. 6801 of *LNCS*, pp. 25–36. 167

[Jac01] JACCARD P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles 37* (1901), 547–579. 40, 57

[JTST10] JAIN A., THORMÄHLEN T., SEIDEL H.-P., THEOBALT C.: Moviereshape: tracking and reshaping of humans in videos. *ACM Transactions on Graphics (SIGGRAPH Asia 2010) 29*, 6 (2010), 148:(1–9). 59

[Jun12] JUNG F.: *Automatische Detektion von Organen in CT-Bildern auf Basis des Viola-Jones-Verfahrens*. Master's thesis, Technische Universität Darmstadt, 2012. 127

[Kal96] KALMAN D.: A singularly valuable decomposition: The SVD of a matrix. *The College Mathematics Journal 27*, 1 (1996), 2–23. 18

[KBW11] KIRSCHNER M., BECKER M., WESARG S.: 3D active shape model segmentation with nonlinear shape priors. In *Medical Image Computing and Computer-Assisted Intervention* (2011), Fichtinger G., Martel A., Peters T., (Eds.), vol. 6892 of *Lecture Notes in Computer Science*, Springer, pp. 492–499. 152

[KGWB11] KIRSCHNER M., GOLLMER S., WESARG S., BUZUG T.: Optimal initialization for 3D correspondence optimization: An evaluation study. In *Information Processing in Medical Imaging* (2011), Székely G., Hahn H., (Eds.), vol. 6801 of *LNCS*, Springer, pp. 308–319. 41, 63, 70, 99

[KLH*13] KAINMUELLER D., LAMECKER H., HELLER M., WEBER B., HEGE H.-C., ZACHOW S.: Omnidirectional displacements for deformable surfaces. *Medical Image Analysis 17*, 4 (2013), 429–441. 46, 49, 54, 166

[KLL07] KAINMUELLER D., LANGE T., LAMECKER H.: Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In *MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge* (2007), Heimann T., et al., (Eds.), pp. 109–116. 4, 42, 46, 48, 53, 60, 132, 143

[KLZH09] KAINMUELLER D., LAMECKER H., ZACHOW S., HEGE H.-C.: An articulated statistical shape model for accurate hip joint segmentation. In *IEEE Engineering in Medicine and Biology Society* (2009), pp. 6345–6351. 24

[KOE*09] KLINDER T., OSTERMANN J., EHM M., FRANZ A., KNESER R., LORENZ C.: Automated model-based vertebra detection, identification, and segmentation in CT images. *Medical Image Analysis 13*, 3 (2009), 471–482. 46, 52, 59, 144, 150

[KPL*03] KAUS M., PEKAR V., LORENZ C., TRUYEN R., LOBREGT S., WEESE J.: Automated 3-D PDM construction from segmented images using deformable models. *IEEE Transactions on Medical Imaging 22*, 8 (2003), 1005–1013. 29, 31

[KSG99] KELEMEN A., SZÉKELY G., GERIG G.: Elastic model-based segmentation of 3-D neuroradiological data sets. *IEEE Transactions on Medical Imaging 18*, 10 (1999), 828–839. 29, 30, 31, 42, 47, 99

[KT98] KOTCHEFF A. C. W., TAYLOR C. J.: Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis 2* (1998), 303–314. 33, 34, 35

[KTL*08] KOIKKALAINEN J., TÖLLI T., LAUERMA K., ANTILA K., MATTILA E., LILJA M., LÖTJÖNEN J.: Methods of artificial enlargement of the training set for statistical shape models. *IEEE Transactions on Medical Imaging 27*, 11 (2008), 1643–1654. 19, 60

[KvdHL*08] KLEIN S., VAN DER HEIDE U. A., LIPS I. M., VAN VULPEN M., STARING M., PLUIM J. P. W.: Automatic segmentation of the prostate in 3D MR images by atlas matching using localized mutual information. *Medical Physics 35*, 4 (2008), 1407–1417. 152, 153

[KW10] KIRSCHNER M., WESARG S.: Construction of groupwise consistent shape parameterizations by propagation. In *SPIE Medical Imaging 2010: Image Processing* (2010), Dawant B. M., Haynor D. R., (Eds.), vol. 7623. 99

[KWL*08] KLINDER T., WOLZ R., LORENZ C., FRANZ A., OSTERMANN J.: Spine segmentation using articulated shape models. In *Medical Image Computing and Computer-Assisted Intervention* (2008), Metaxas D. N., Axel L., Fichtinger G., Székely G., (Eds.), vol. 5241 of *LNCS*, Springer, pp. 227–234. 24, 25

[KWT88] KASS M., WITKIN A. P., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (1988), 321–331. 3

[LAV09] LÜTHI M., ALBRECHT T., VETTER T.: Building shape models from lousy data. In *Medical Image Computing and Computer-Assisted Intervention* (2009), Yang G.-Z., Hawkes D., Rueckert D., Noble A., Taylor C., (Eds.), vol. 5762 of *LNCS*, Springer, pp. 1–8. 18

[LB07] LEUNG E. K. Y., BOSCH J. G.: Localized shape variations for classifying wall motion in echocardiograms. In *Medical Image Computing and Computer-Assisted Intervention* (2007), Ayache N., Ourselin S., Maeder A. J., (Eds.), vol. 4791 of *LNCS*, pp. 52–59. 20

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Computer Graphics 21*, 4 (1987), 163–169. 26, 30, 96

[LCJ*12] LU C., CHELIKANI S., JAFFRAY D., MILOSEVIC M., STAIB L., DUNCAN J.: Simultaneous nonrigid registration, segmentation, and tumor detection in MRI guided cervical cancer radiation therapy. *IEEE Transactions on Medical Imaging 31*, 6 (2012), 1213–1227. 1

[LGF00] LEVENTON M., GRIMSON W., FAUGERAS O.: Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition* (2000), vol. 1, pp. 316–323. 10, 13

[LK00] LORENZ C., KRAHNSTOEVER N.: Generation of point-based 3D statistical shape models for anatomical objects. *Computer Vision and Image Understanding 77* (2000), 175–181. 29, 30, 32

[LKK*04] LÖTJÖNEN J., KIVISTÖ S., KOIKKALAINEN J., SMUTEK D., LAUERMA K.: Statistical shape model of atria, ventricles and epicardium from short- and long-axis MR images. *Medical Image Analysis 8*, 3 (2004), 371–386. 20, 21

[LLS02] LAMECKER H., LANGE T., SEEBASS M.: A statistical shape model for the liver. In *Medical Image Computing and Computer-Assisted Intervention* (2002), Dohi T., Kikinis R., (Eds.), vol. 2489 of *LNCS*, Springer, pp. 422–427. 29, 30

[LMY07] LEKADIR K., MERRIFIELD R. D., YANG G.-Z.: Outlier detection and handling for robust 3-D active shape models search. *IEEE Transactions on Medical Imaging 26*, 2 (2007), 212–222. 49

[LN89] LIU D. C., NOCEDAL J.: On the limited memory BFGS method for large scale optimization. *Mathematical Programming B 45* (1989), 503–528. 88, 121, 169

[LSHD04] LAMECKER H., SEEBASS M., HEGE H., DEUFLHARD P.: A 3D statistical shape model of the pelvic bone for segmentation. In *SPIE Medical Imaging: Image Processing* (2004), Fitzpatrick J. M., Sonka M., (Eds.), vol. 5370. 30, 46

[LXAG09] LI C., XU C., ANDERSON A. W., GORE J. C.: MRI tissue classification and bias field estimation based on coherent local intensity clustering: A unified energy minimization framework. In *Information Processing in Medical Imaging* (2009), Prince J. L., Pham D. L., Myers K. J., (Eds.), vol. 5636 of *LNCS*, pp. 288–299. 153, 154, 159

[LZZ*08] LING H., ZHOU S., ZHENG Y., GEORGESCU B., SUEHLING M., COMANICIU D.: Hierarchical, learning-based automatic liver segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2008), pp. 1–8. 44

[MB04] MATTHEWS I., BAKER S.: Active appearance models revisited. *International Journal of Computer Vision 60*, 2 (2004), 135–164. 55

[MDT08] MARTIN S., DAANEN V., TROCCAZ J.: Atlas-based prostate segmentation using an hybrid registration. *International Journal of Computer-Assisted Radiology and Surgery 3* (2008), 485–492. 152, 153

[MDW08] MUNSELL B., DALAL P., WANG S.: Evaluating shape correspondence for statistical shape analysis: A benchmark study. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 11 (2008), 2023–2039. 40

[MFR*08] MEI L., FIGL M., RUECKERT D., DARZI A., EDWARDS P.: Sample sufficiency and number of modes to retain in statistical shape modelling. In *Medical Image Computing and Computer-Assisted Intervention* (2008), Metaxas D., Axel L., Fichtinger G., Székely G., (Eds.), vol. 5241 of *LNCS*, Springer, pp. 425–433. 14

[MP97] MOGHADDAM B., PENTLAND A.: Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19* (1997), 696–710. 16, 25, 115, 125

[MPL*09] MAKNI N., PUECH P., LOPES R., DEWALLE A., COLOT O., BETROUNI N.: Combining a deformable model and a probabilistic framework for an automatic 3D segmentation of prostate on MRI. *International Journal of Computer-Assisted Radiology and Surgery 4* (2009), 181–188. 53, 152, 153

[MS10] MYRONENKO A., SONG X.: Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 12 (2010), 2262–2275. 85, 88

[MSW*11] MONTILLO A., SHOTTON J., WINN J., IGLESIAS J. E., METAXAS D., CRIMINISI A.: Entangled decision forests and their application for semantic segmentation of CT images. In *Information Processing in Medical Imaging* (2011), Székely G., Hahn H., (Eds.), vol. 6801 of *LNCS*, pp. 184–196. 167

[MTC02] MEINZER H.-P., THORN M., CÁRDENAS C. E.: Computerized planning of liver surgery–an overview. *Computers & Graphics 26*, 4 (2002), 569–576. 1, 132

[MTSW12] MUNSELL B. C., TEMLYAKOV A., STYNER M., WANG S.: Pre-organizing shape instances for landmark-based shape correspondence. *International Journal of Computer Vision 97*, 2 (2012), 210–228. 2. 32, 33, 89, 90, 91, 95, 101

[ODED12] OU Y., DOSHI J., ERUS G., DAVATZIKOS C.: Multi-atlas segmentation of the prostate: A zooming process with robust registration and atlas selection. In *PROMISE12 – MICCAI 2012 Grand Challenge on Prostate MR Image Segmentation* (2012). 153

[ONC*09] OGUZ I., NIETHAMMER M., CATES J. E., WHITAKER R. T., FLETCHER P. T., VACHET C., STYNER M.: Cortical correspondence with probabilistic fiber connectivity. In *Information Processing in Medical Imaging* (2009), Prince J. L., Pham D. L., Myers K. J., (Eds.), vol. 5636 of *LNCS*, pp. 651–663. 35

[OS88] OSHER S., SETHIAN J.: Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics 79*, 1 (1988), 12–49. 10

[PFB*07] POHL K. M., FISHER J., BOUIX S., SHENTON M., MCCARLEY R. W., GRIMSON W. E. L., KIKINIS R., WELLS W. M.: Using the logarithm of odds to define a vector space on probabilistic atlases. *Medical Image Analysis 11*, 5 (2007), 465–477. 10

[PFJ*03] PIZER S. M., FLETCHER P., JOSHI S., THALL A., CHEN J. Z., FRIDMAN Y., FRITSCH D. S., GASH A. G., GLOTZER J. M., JIROUTEK M. R., LU C., MULLER K. E., TRACTON G., YUSHKEVICH P., CHANEY E. L.: Deformable m-reps for 3D medical image segmentation. *International Journal of Computer Vision 55*, 2–3 (2003), 85–106. 11

[PFY*99] PIZER S. M., FRITSCH D. S., YUSHKEVICH P. A., JOHNSON V. E., CHANEY E. L.: Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging 18*, 10 (1999), 851–865. 11

[PH03a] PAULSEN R., HILGER K.: Shape modelling using markov random field restoration of point correspondences. In *Information Processing in Medical Imaging* (2003), Taylor C., Noble J., (Eds.), vol. 2732 of *LNCS*, Springer, pp. 1–12. 29, 30, 32

[PH03b] PRAUN E., HOPPE H.: Spherical parametrization and remeshing. *ACM Transactions on Graphics 22*, 3 (2003), 340–349. 65

[Pro] http://promise12.grand-challenge.org/. 4, 12, 172

[PTVF92] PRESS W., TEUKOLSKY S., VETTERLING W., FLANNERY B.: *Numerical recipes in C*, 2nd ed. Cambridge University Press, 1992. 66, 68

[RAR08] RAO A., ALJABAR P., RUECKERT D.: Hierarchical statistical shape analysis and prediction of sub-cortical brain structures. *Medical Image Analysis 12*, 1 (2008), 55–68. 25

[RBP01] ROK B., BOŠTJAN L., PERNUŠ F.: Segmenting articulated structures by hierarchical statistical modeling of shape, appearance, and topology. In *Medical Image Computing and Computer-Assisted Intervention* (2001), Niessen W., Viergever M., (Eds.), vol. 2208 of *LNCS*, Springer, pp. 499–506. 24, 25

[RC05] ROUSSON M., CREMERS D.: Efficient kernel density estimation of shape and intensity priors for level set segmentation. In *Medical Image Computing and Computer-Assisted Intervention* (2005), Duncan J. S., Gerig G., (Eds.), vol. 3750 of *LNCS*, Springer, pp. 757–764. 10

[RCS*11] RUPPERTSHOFEN H., CRISTIAN L., SCHMIDT S., BEYERLEIN P., SALAH Z., ROSE G., SCHRAMM H.: Discriminative generalized hough transform for localization of joints in the lower extremities. *Computer Science – Research and Development 26*, 1-2 (2011), 97–105. 43

[RFS03] RUECKERT D., FRANGI A., SCHNABEL J.: Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration. *IEEE Transactions on Medical Imaging 22*, 8 (2003), 1014–1025. 20

[RG02] ROGERS M., GRAHAM J.: Robust active shape model search. In *European Conference on Computer Vision* (2002), Heyden A., Sparr G., Nielsen M., Johansen P., (Eds.), vol. 2353 of *LNCS*, Springer, pp. 517–530. 49, 50

[RGP99] ROMDHANI S., GONG S., PSARROU A.: A multi-view nonlinear active shape model using kernel PCA. In *British Machine Vision Conference* (1999), Pridmore T. P., Elliman D., (Eds.), pp. 483–492. 23, 124

[RST*07] RAJAMANI K., STYNER M., TALIB H., ZHENG G., NOLTE L.-P., GONZÁLEZ BALLESTER M. Á.: Statistical deformable bone models for robust 3D surface extrapolation from sparse data. *Medical Image Analysis 11*, 2 (2007), 99–109. 59

[SBKW13] STEGER S., BOZOGLU N., KUIJPER A., WESARG S.: Application of radial ray based segmentation to cervical lymph nodes in CT images. *IEEE Transactions on Medical Imaging* (2013). online preprint. 1

[SBZ*09] SEIFERT S., BARBU A., ZHOU S. K., LIU D., FEULNER J., HUBER M., SUEHLING M., CAVALLARO A., COMANICIU D.: Hierarchical parsing and semantic navigation of full body CT data. In *SPIE Medical Imaging 2009: Image Processing* (2009), Pluim J. P. W., Dawant B. M., (Eds.), vol. 7259. 44

[SCT*97] SOZOU P., COOTES T., TAYLOR C., DI MAURO E., LANITIS A.: Non-linear point distribution modelling using a multi-layer perceptron. *Image and Vision Computing 15* (1997), 457–463. 21

[SCTDM95] SOZOU P., COOTES T., TAYLOR C., DI MAURO E.: Non-linear generalisation of point distribution models using polynomial regression. *Image and Vision Computing 13* (1995), 451–458. 21

[SGJP03] STYNER M., GERIG G., JOSHI S., PIZER S. M.: Automatic and robust computation of 3D medial models incorporating object variability. *International Journal of Computer Vision 55*, 2–3 (2003), 107–122. 11

[SHM07] SCHÖNIG M., HEIMANN T., MEINZER H.-P.: Parametrisierung geschlossener Oberflächen für die Erzeugung von 3D-Formmodellen. In *Bildverarbeitung für die Medizin* (2007), Horsch A., Deserno T. M., Handels H., Meinzer H.-P., Tolxdorff T., (Eds.), pp. 394–398. 64

[Sho85] SHOEMAKE K.: Animating rotation with quaternion curves. *SIGGRAPH Computer Graphics 19*, 3 (1985), 245–254. 71

[Ski] http://www.ski10.org. 4, 12

[SKW12] STEGER S., KIRSCHNER M., WESARG S.: Articulated atlas for segmentation of the skeleton from head amp; neck CT datasets. In *IEEE International Symposium on Biomedical Imaging* (2012), pp. 1256–1259. 25

[SLPG04] STYNER M., LIEBERMAN J., PANTAZIS D., GERIG G.: Boundary and medial shape analysis of the hippocampus in schizophrenia. *Medical Image Analysis 8*, 3 (2004), 197–203. 59

[SOHG08] STYNER M., OGUZ I., HEIMANN T., GERIG G.: Minimum description length with local geometry. In *IEEE International Symposium on Biomedical Imaging* (2008), pp. 1283–1286. 35

[SP98] SUNG K.-K., POGGIO T.: Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 1 (1998), 39–51. 128

[SP05] STEGMANN M. B., PEDERSEN D.: Bi-temporal 3D active appearance models with applications to unsupervised ejection fraction estimation. In *SPIE Medical Imaging: Image Processing* (2005), Fitzpatrick J. M., Reinhardt J. M., (Eds.), vol. 5747, pp. 336–350. 55

[SP08] SIDDIQI K., PIZER S. M.: *Medial Representations: Mathematics, Algorithms and Applications*, 1 ed. Springer, 2008. 11

[SRN*03] STYNER M., RAJAMANI K., NOLTE L.-P., ZSEMLYE G., SZÉKELY G., TAYLOR C. J., DAVIES R. H.: Evaluation of 3D correspondence methods for model building. In *Information Processing in Medical Imaging*, Taylor C. J., Noble J., (Eds.), vol. 2732 of *LNCS*. Springer, 2003, pp. 63–75. 41, 97

[SRR*07] Sjöstrand K., Rostrup E., Ryberg C., Larsen R., Studholme C., Baezner H., Ferro J., Fazekas F., Pantoni L., Inzitari D., Waldemar G.: Sparse decomposition and modeling of anatomical shape variation. *IEEE Transactions on Medical Imaging 26*, 12 (2007), 1625–1635. 20

[SSL06] Stegmann M. B., Sjöstrand K., Larsen R.: Sparse modeling of landmark and texture variability using the orthomax criterion. In *SPIE Medical Imaging: Image Processing* (2006), Reinhardt J. M., Pluim J. P. W., (Eds.), vol. 6144. 20

[SSM98] Schölkopf B., Smola A. J., Müller K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10*, 5 (1998), 1299–1319. 22

[SÜF*04] Suinesiaputra A., Üzümcü M., Frangi A. F., Kaandorp T. A. M., Reiber J. H. C., Lelieveldt B. P. F.: Detecting regional abnormal cardiac contraction in short-axis mr images using independent component analysis. In *Medical Image Computing and Computer-Assisted Intervention* (2004), Barillot C., Haynor D. R., Hellier P., (Eds.), vol. 3216 of *LNCS*, pp. 737–744. 20

[SZL92] Schroeder W. J., Zarge J. A., Lorensen W. E.: Decimation of triangle meshes. *SIGGRAPH Computer Graphics 26*, 2 (1992), 65–70. 30

[TCR*08] Toth R., Chappelow J., Rosen M., Pungavkar S., Kalyanpur A., Madabhushi A.: Multi-attribute non-initializing texture reconstruction based active shape model (MANTRA). In *Medical Image Computing And Computer-Assisted Intervention* (2008), Metaxas D. N., A. L., Fichtinger G., Székely G., (Eds.), vol. 5241 of *LNCS*, pp. 653–661. 47

[TDR*09] Toth R., Doyle S., Rosen M., Kalyanpur A., Pungavkar S., Bloch B. N., Genega E., Rofsky N., Lenkinski R., Madabhushi A.: WERITAS: weighted ensemble of regional image textures for ASM segmentation. In *SPIE Medical Imaging: Image Processing* (2009), Pluim J. P. W., Dawant B. M., (Eds.), vol. 7259. 47

[Tho03] Thodberg H. H.: Minimum description length shape and appearance models. In *Information Processing in Medical Imaging* (2003), Taylor C. J., Noble J. A., (Eds.), vol. 2732 of *LNCS*, pp. 51–62. 34, 38

[TICB09] Tejos C., Irarrazaval P., Cárdenas-Blanco A.: Simplex mesh diffusion snakes: Integrating 2D and 3D deformable models and statistical shape knowledge in ăvariational framework. *International Journal of Computer Vision 85* (2009), 19–34. 57

[TM12] Toth R., Madabhushi A.: Multi-feature landmark-free active appearance models: Application to prostate MRI segmentation. *IEEE Transactions on Medical Imaging 31*, 8 (2012), 1638–1650. 11, 153

[TT01] Twining C. J., Taylor C. J.: Kernel principal component analysis and the construction of non-linear active shape models. In *British Machine Vision Conference* (2001), Cootes T., Taylor C., (Eds.). 23, 53, 124, 125

[TT11] TWINING C., TAYLOR C. J.: Specificity: A graph-based estimator of divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 12 (2011), 2492–2504. 39

[TTR*11] TOTH R., TIWARI P., ROSEN M., REED G., KURHANEWICZ J., KALYANPUR A., PUNGAVKAR S., MADABHUSHI A.: A magnetic resonance spectroscopy driven initialization scheme for active shape model based prostate segmentation. *Medical Image Analysis 15*, 2 (2011), 214–225. 42

[Tu05] TU Z.: Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *IEEE International Conference on Computer Vision* (2005), IEEE Computer Society, pp. 1589–1596. 43

[Tut63] TUTTE W.: How to draw a graph. *Proceedings of the London Mathematical Society 13*, 3 (1963), 743–768. 74, 79, 80

[TZB*06] TU Z., ZHOU X. S., BOGONI L., BARBU A., COMANICIU D.: Probabilistic 3D polyp detection in CT images: The role of sample alignment. In *IEEE Conference on Computer Vision and Pattern Recognition* (2006), vol. 2, pp. 1544–1551. 43

[UFRL03] ÜZÜMCÜ M., FRANGI A. F., REIBER J. H., LELIEVELDT B. P.: Independent component analysis in statistical shape models. In *SPIE Medical Imaging 2003: Image Processing* (2003), Sonka M., Fitzpatrick J. M., (Eds.), vol. 5032, pp. 375–383. 20

[UFS*03] ÜZÜMCÜ M., FRANGI A. F., SONKA M., REIBER J. H. C., LELIEVELDT B. P. F.: ICA vs. PCA active appearance models: Application to cardiac MR segmentation. In *Medical Image Computing and Computer-Assisted Intervention* (2003), Ellis R. E., Peters T. M., (Eds.), vol. 2878 of *LNCS*, Springer, pp. 451–458. 21

[UNSF08] UNAL G., NAIN D., SLABAUGH G., FANG T.: Customized design of hearing aids using statistical shape learning. In *Medical Image Computing and Computer-Assisted Intervention* (2008), Metaxas D., Axel L., Fichtinger G., Székely G., (Eds.), vol. 5241 of *LNCS*, Springer, pp. 518–526. 59

[VBG*08] VISWANATH S., BLOCH B., GENEGA E., ROFSKY N., LENKINSKI R., CHAPPELOW J., TOTH R., MADABHUSHI A.: A comprehensive segmentation, registration, and cancer detection scheme on 3 tesla in vivo prostate DCE-MRI. In *Medical Image Computing and Computer-Assisted Intervention* (2008), Metaxas D., Axel L., Fichtinger G., Székely G., (Eds.), vol. 5241 of *LNCS*, Springer, pp. 662–669. 152

[VdBA*04] VOS F., DE BRUIN P., AUBEL J., STREEKSTRA G., MAAS M., VAN VLIET L., VOSSEPOEL A.: A statistical shape model without using landmarks. In *International Conference on Pattern Recognition* (2004), vol. 3, pp. 714–717. 29, 31

[VGB12] VINCENT G., GUILLARD G., BOWES M.: Fully automatic segmentation of the prostate using active appearance models. In *PROMISE12 - MICCAI 2012 Grand Challenge on Prostate MR Image Segmentation* (2012). 153, 157 [vGFS*02] van Ginneken B., Frangi A. F., Staal J., ter Haar Romeny B. M., Viergever M. A.: Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging 21*, 8 (2002), 924–933. 47

[VJ01] Viola P., Jones M.: Robust real-time object detection. *International Journal of Computer Vision 57*, 2 (2001), 137–154. 43, 127, 128, 147

[vKZHCO11] van Kaick O., Zhang H., Hamarneh G., Cohen-Or D.: A survery on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707. 26

[VPL07] Vovk U., Pernuš F., Likar B.: A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Transactions on Medical Imaging 26*, 3 (2007), 405–421. 153

[WKB*12] Wesarg S., Kirschner M., Becker M., Erdt M., Kafchitsas K., Khan M.: Dual-energy CT-based assessment of the trabecular bone in vertebrae. *Methods of Information in Medicine 51*, 5 (2012), 398–405. 144

[WKL*01] Weese J., Kaus M., Lorenz C., Lobregt S., Truyen R., Pekar V.: Shape constrained deformable models for 3D medical image segmentation. In *Information Processing in Medical Imaging* (2001), Insana M. F., Leahy R. M., (Eds.), vol. 2082 of *LNCS*, Springer, pp. 380–387. 4, 46, 48, 51, 52, 54, 59, 60, 122, 123

[WLEH*10] Wang L., Lekadir K., EI-Hamamsy I., Yacoub M., Yang G.-Z.: Subject specific shape modeling with incremental mixture models. In *International Workshop on Medical Imaging and Augmented Reality* (2010), Liao H., Edwards P., Pan X., Fan Y., Yang G.-Z., (Eds.), vol. 6326 of *LNCS*, Springer, pp. 21–30. 22

[WS00] Wang Y., Staib L.: Boundary finding with prior shape and smoothness models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 7 (2000), 738–743. 19, 54

[WSH09] Wimmer A., Soza G., Hornegger J.: A generic probabilistic active shape model for organ segmentation. In *Medical Image Computing and Computer-Assisted Intervention* (2009), Yang G.-Z., Hawkes D. J., Rueckert D., Noble J. A., Taylor C. J., (Eds.), vol. 5762 of *LNCS*, Springer, pp. 26–33. 10, 11, 43, 132

[WZW04] Warfield S. K., Zou K. H., Wells W. M.: Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging 23*, 7 (2004), 903–921. 1, 57

[YOT*09] Yokota F., Okada T., Takao M., Sugano N., Tada Y., Sato Y.: Automated segmentation of the femur and pelvis from 3D CT data of diseased hip using hierarchical statistical shape model of joint structure. In *Medical Image Computing and Computer-Assisted Intervention* (2009), Yang G.-Z., Hawkes D., Rueckert D., Noble A., Taylor C., (Eds.), vol. 5762 of *LNCS*, Springer, pp. 811–818. 24, 25

[ZAT05] ZHAO Z., AYLWARD S., TEOH E.: A novel 3D partitioned active shape model for segmentation of brain MR images. In *Medical Image Computing and Computer-Assisted Intervention* (2005), Duncan J., Gerig G., (Eds.), vol. 3749 of *LNCS*, Springer, pp. 221–228. 19

[ZBG*08] ZHENG Y., BARBU A., GEORGESCU B., SCHEUERING M., COMANICIU D.: Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE Transactions on Medical Imaging 27*, 11 (2008), 1668–1681. 44

[ZGL*09] ZHENG Y., GEORGESCU B., LING H., ZHOU S., SCHEUERING M., COMANICIU D.: Constrained marginal space learning for efficient 3D anatomical structure detection in medical images. In *IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 194–201. 44

[ZT08] ZHAO Z., TEOH E. K.: A new scheme for automated 3D PDM construction using deformable models. *Image and Vision Computing 26* (2008), 275–288. 33