

3D Statistical Shape Models of Human Bones

their Construction using a Finite Element Registration Algorithm,
Formulation on Hilbert Spaces,
and Application to Medical Image Analysis

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Thomas Albrecht
aus Deutschland

Basel, 2011

Originaldokument gespeichert auf dem Dokumentenserver der Universität Basel edoc.unibas.ch



Dieses Werk ist unter dem Vertrag "Creative Commons Namensnennung – Keine kommerzielle Nutzung – Keine Bearbeitung 2.5 Schweiz" lizenziert. Die vollständige Lizenz kann unter <http://creativecommons.org/licenses/by-nc-nd/2.5/ch/legalcode.de> eingesehen werden.



Attribution-NonCommercial-NoDerivs 2.5 Switzerland (CC BY-NC-ND 2.5)¹

You are free:

Ⓒ to copy, distribute, display, and perform the work

Under the following conditions:

- Ⓘ **Attribution** – You must give the original author credit.
- Ⓝ **Non-Commercial** – You may not use this work for commercial purposes.
- Ⓔ **No Derivative Works** – You may not alter, transform, or build upon this work.

With the understanding that:

Waiver – Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain – Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights – In no way are any of the following rights affected by the license: Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations; The author's moral rights; Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice – For any reuse or distribution, you must make clear to others the licence terms of this work.

¹ This Commons Deed is a human-readable summary of the Legal Code (the full licence). You can get the full licence by visiting <http://creativecommons.org/licenses/by-nc-nd/2.5/ch/legalcode.de>.

Disclaimer – The Commons Deed is not a license. It is simply a handy reference for understanding the Legal Code (the full licence) – it is a human-readable expression of some of its key terms. Think of it as the user-friendly interface to the Legal Code beneath. This Deed itself has no legal value, and its contents do not appear in the actual license.

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter
Prof. Dr. Thomas Brox, Universität Freiburg, Korreferent

Basel, den 14.12.2010

Prof. Dr. Martin Spiess, Dekan

Abstract

Statistical shape models have become a widely used tool in computer vision and medical image analysis. They are constructed from a representative set of example shapes and represent the normal shape variations of a class of objects, in our case of human bones. The foundation of statistical shape models is the concept of correspondence. In order to draw meaningful statistical conclusions and to build a generative model from the example shapes, we should compare and relate only corresponding parts of the shape. The task of establishing correspondence between shapes and images is known as the registration problem and is one of the fundamental problems of computer vision. To approximate a solution of the registration problem for our bone shapes, we propose a new registration algorithm, which is formulated as a continuous minimization problem, whose solution is sought with a state of the art finite element method.

Once the shapes have been brought into correspondence, a statistical shape model can be built. We present a formulation of the shape model on general Hilbert spaces, which incorporates all associated models which can be constructed in a similar way, like models of shape, color, intensity, deformations etc. Which of these models is used depends only on the choice of the Hilbert space. Because this includes the choice between continuously defined models and models based on any kind of discretization method, we can easily integrate the statistical model into our registration method and its finite element discretization. This inclusion of class-specific prior knowledge into makes the registration more robust against outliers and damaged data sets.

Finally, we show how the statistical models can be applied to a number of practical problems from medical image analysis and surgery planning, like the fitting of the model to novel shapes or images, the design of optimized medical implants or the automatic repositioning of fractured bones.

Contents

Abstract	5
Notation	9
1 Introduction	11
1.1 Shape Models	13
1.1.1 Correspondence	14
1.1.2 Registration	15
1.1.3 Morphable Model	17
1.2 Outline and Contributions	17
1.3 Prior Work	19
2 Data Acquisition and Preprocessing	21
2.1 CT Scanning	21
2.2 Segmentation	22
2.3 Pre-Alignment	25
3 Registration	27
3.0.1 Prior Work	28
3.1 Registration Method	29
3.1.1 Level Set Representation	29
3.1.2 Function Spaces	31
3.1.3 Distance Term	31
3.1.4 Robust Distance Measures	32
3.1.5 Curvature Guided Registration	32
3.1.6 Regularization Term	35
3.1.7 Full Registration Functional	37
3.2 Minimization Strategy	37
3.2.1 Existence of Minimizers of the Registration Functional	38
3.2.2 Euler-Lagrange Equations	40
3.2.3 Derivatives	41
3.2.4 Strong Derivative	42
3.3 Discretization	42
3.3.1 Time Discretization	42
3.3.2 Finite Element Space Discretization	44
3.3.3 Local Discontinuous Galerkin Finite Element Discretization	46
3.3.4 Implementational Details	49
3.4 Registration Results	50

3.4.1	3D Surface Registration	51
3.4.2	Curvature Term	53
3.4.3	Volume Preservation	54
3.4.4	Parameters	56
4	The Statistical Shape Model	57
4.1	Shape Models on Hilbert Spaces	61
4.1.1	Dimensionality Reduction	65
4.1.2	The Normal Distribution	65
4.1.3	Functional PCA	68
4.2	Visualization	70
4.3	Discrete Models	72
4.3.1	Intensity Model	75
4.4	Dealing with \mathcal{M}^+	77
4.4.1	Shrinkage Estimation	78
4.4.2	A Note on Noise	80
4.4.3	PPCA	81
5	Applications	83
5.1	Statistically Regularized Registration	83
5.1.1	Registration Results	85
5.2	Statistical Shape Model Fitting	87
5.2.1	Fitting with known Correspondence	90
5.2.2	Reconstruction of Partial Shapes	91
5.2.3	Flexibility Model	94
5.2.4	Visualizing the Flexibility	95
5.2.5	Limitations of the Method	97
5.2.6	Model Fitting without known Correspondence	98
5.2.7	Generic Outside Model	101
5.2.8	Mumford Shah Model	102
5.2.9	Implementation	104
5.2.10	Segmentation Results	105
5.3	Building Shape Models from Problematic Data	106
5.4	Orthopedic Implant Design	108
5.5	Predicting Faces from Skulls	112
5.6	Visualizing the Density of the Subchondral Bone	113
5.7	Fracture Reduction	116
5.7.1	Open Issues	119
6	Discussion	123
	Bibliography	125
	Curriculum Vitae	131

Notation

\mathbb{R}	Real numbers
\mathbb{R}^+	Positive numbers
\mathbb{N}	Natural numbers $\{1, 2, \dots\}$
$\text{span}\{\rho_1, \dots, \rho_m\}$	Linear span of the vectors ρ_1, \dots, ρ_m
img	Image of an operator or matrix
ker	Kernel of an operator or matrix
\circ	Composition (of functions, operators etc.)
\cong	Isomorphic
$L^2(\Omega)$	The space of all square-integrable functions $u : \Omega \rightarrow \mathbb{R}$
$L^2(\Omega, \mathbb{R}^d)$	The space of all square-integrable functions $u : \Omega \rightarrow \mathbb{R}^d$
$H^1(\Omega, \mathbb{R}^d) = H^{1,2}(\Omega, \mathbb{R}^d)$	The space of all functions in $L^2(\Omega, \mathbb{R}^d)$ with weak first derivative
$H^2(\Omega, \mathbb{R}^d) = H^{2,2}(\Omega, \mathbb{R}^d)$	The space of all functions in $L^2(\Omega, \mathbb{R}^d)$ with weak second derivative
$\int_{\Omega} f(x) dx$	Lebesgue Integral over Ω
$\int_{\Gamma} f(x) do(x)$	Surface Integral over Γ
$\ \cdot\ _{L^2(\Omega, \mathbb{R}^d)}$	$L^2(\Omega, \mathbb{R}^d)$ norm
$\ \cdot\ $	Euclidean norm
\mathcal{H}	Hilbert space
$\ \cdot\ _{\mathcal{H}}$	Hilbert space norm
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Hilbert space scalar product
\mathbf{I}_N	Identity matrix in \mathbb{R}^N
id	The identity map
$\mathcal{N}(\mu, \Sigma)$	Normal distribution with mean μ and covariance matrix Σ
$\mathcal{N}(\mu, \mathcal{C})$	Normal distribution with mean μ and covariance operator \mathcal{C}
n	The number of training examples
m	The number of nonzero eigenvalues of the covariance operator
DOF	Degree of freedom

1 Introduction

Since the invention of photography and radiography, the number of images acquired both in everyday life and in the scientific and medical domain has risen steadily. More recent inventions like computed tomography (CT), magnetic resonance imaging (MRI), and digital photography, and their growing affordability, have accelerated this process dramatically.

The simultaneous advance in computer science has allowed the development of methods to support humans in the analysis of these images. For instance, segmentation algorithms aim at extracting and presenting relevant information from an image. Algorithms for computer aided diagnostics try to automatically detect suspicious regions in a medical image. Increasingly, these computer assisted methods are becoming a necessity rather than a convenience, as without them it becomes more and more difficult to analyze and handle the sheer amount of image data that is constantly being acquired.

There are many tasks in image analysis which computers can carry out more efficiently than humans. These include for instance calculations on all the pixels in an image, or measuring the volume of regions within an image. For instance, in the images in Figure 1.1, a computer takes only milliseconds to compute the mean and standard deviation of the color of all the pixels in the image, whereas it would take a human days to calculate these numbers by hand.

In many other tasks however, humans are by far superior to computers, especially when we move from monotonous tasks like pixel counting to actually seeing and understanding images. For instance, it will most likely still take decades of research before a computer can give any kind of sensible answer to the question: “What do you see in these images?”, whereas a human takes only seconds to regard the image and answers something like: “I see a man jumping his bike off a ramp. It looks like he fell and broke his bone.”

Trying to understand how humans are able to interpret these images this efficiently is an active area of research for itself. A part of an answer to this question is certainly that we as humans can rely on the experience we have gained in our life. A person who has never seen a bike or an x-ray image will certainly have a harder time at understanding these images. On the other hand, a person with more experience or knowledge will be able to extract more information, such as the exact type of bone fracture or the type of jump the mountain biker is attempting.

The computer science disciplines of computer vision and machine learning develop algorithms that work towards making computers “see” and “understand” images. The most promising approach in this area is to model some kind of “experience” or “knowledge” in the computer, which can then be used in the interpretation of images similar to how a radiologist draws from his experience and knowledge in order to diagnose medical images. Of course, the algorithms of computer vision and machine learning are still very far from being able to imitate the various ways in which humans acquire, remember and use knowledge.

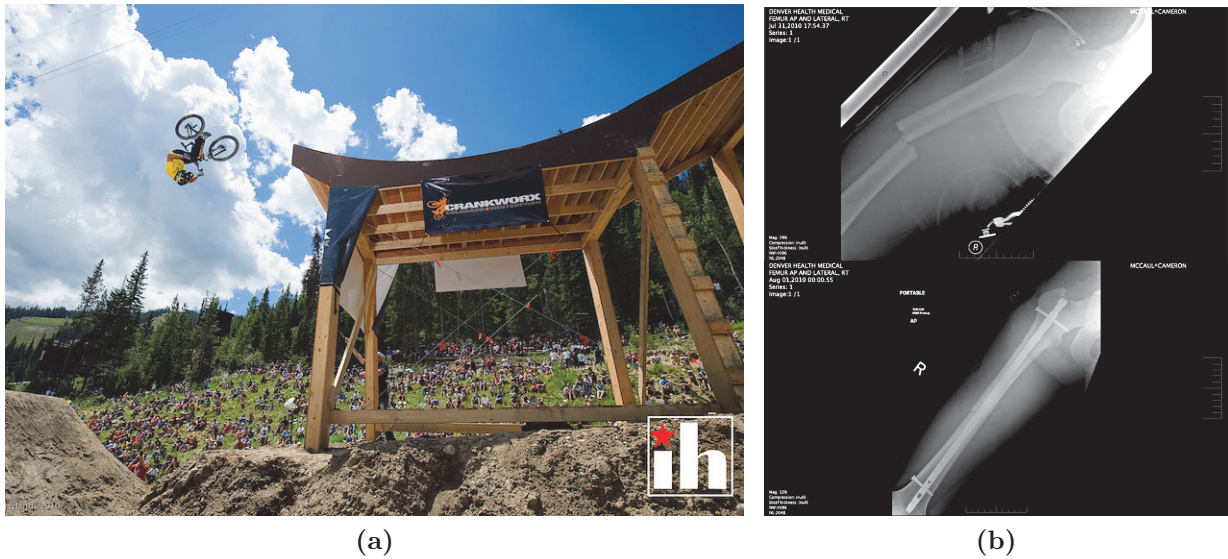


Figure 1.1: (a) shows a mountain biker attempting a front flip. (b) shows radiographs of the femur fracture he sustained in the attempt and its surgical treatment by fixation with a long metal nail [38].

Nevertheless, we can try to mimic a very simple type of learning method: A person can learn a lot about a class of objects by simply looking at a large number of examples. For instance, by looking at a number of femur bones, a person will learn how the bone typically looks like and what variations are common. He forms a mental model of the femur in his mind.

This is what we try to imitate on a computer. From a set of examples, we construct a model of a class of objects. This model represents the knowledge about this class of objects and can be used in the analysis of images that show an object of the class. The models we use in this thesis are *3D shape models*. The most prominent example that started the research for this thesis is the human femur bone. But for many experiments and illustrations, including those in this introduction, we also use the skull, as well as other object classes, like tibiae, faces and hands. Each of these classes require their own model. Of course this makes it impossible to ever acquire the rich knowledge a human possesses, but we can show that for individual object classes, this method can be put to good use. For instance, for the object class of femur bones, we can use a shape model to automatically compute an optimal repositioning of the fragments of a fracture like that in Figure 1.1b. Without computer support, this is surprisingly hard to achieve for humans, resulting in operations in which the bones that are poorly aligned, see [40].

The shape modeling technique we use is based on the 3D Morphable Model [13], which was originally introduced for human faces. Most research in the Graphics and Vision research group at the University of Basel is dedicated to this face model and it is the task of this thesis to apply this modelling method in the medical domain. For this, every method from data acquisition to the applications of the model had to be reevaluated regarding its use with medical data. The 3D Morphable Model showed that it is ineffective to try to

learn the shape of 3D objects from 2D images like photos or the 2D radiographs shown in Figure 1.1b. Instead, we need to acquire examples that show the 3D shape of the bone. Currently, this information is best acquired in the form of CT scans. We will show in Chapter 2 how the 3D shape information can be extracted from these scans. In this thesis we focus on bone models, but these should be seen as an example for the many object classes that can be modeled with 3D shape models.

1.1 Shape Models

The basic idea of a shape model is to represent a class of shapes by combinations of example shapes. As with all statistical methods, the more examples are used, the better the model can represent the class of objects it models. The models we use here are *linear* shape models. This means that they represent a class of shapes as linear combinations of the example shapes. While this seems very straight-forward, it is not instantly clear how these linear combinations can be formed. It is not even clear how we can take the sum of two shapes. An early and rather simple method was introduced as Eigenfaces by Turk and Pentland in [70]. It represents face shapes by images and simply builds a model based on linear combination of these images. Similarly, Leventon et al. [42] represent shapes like human vertebrae by distance images and use linear combinations of these distance images to build a shape model. However, for these methods it is in general not certain that a linear combination represents again a valid shape from the object class. In fact, for more complicated shapes, this concept of shape modeling breaks down completely. In Figure 1.2 a linear combination of only two skulls is computed by means of their distance images. Even this most simple of linear combinations fails to represent a valid skull.

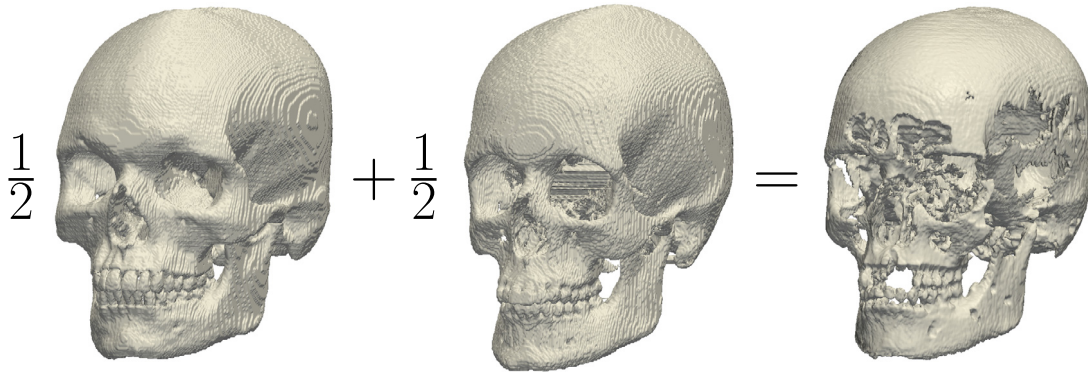


Figure 1.2: Linear combination of two skulls by means of a linear combination of their distance maps as proposed in [42]. The linear combination fails to represent a valid skull.

1.1.1 Correspondence

To construct valid linear combinations of shapes the Morphable Model [13] brings the shapes into *correspondence* before building the shape model. The concept of correspondence is easy to grasp but difficult to define in a mathematically precise way. We consider two parts of a shape to correspond to each other if they share a comparable position or function within their respective shapes. For instance, the teeth in one skull correspond to the teeth in another skull. This concept can be extended to individual points: a point on the corner of one tooth correspond to the point on the corner of the corresponding tooth.

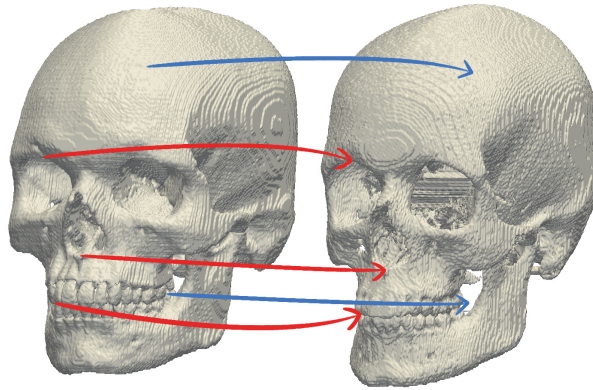


Figure 1.3: Correspondence between two skulls.

While this is a very straight-forward concept, it becomes quite vague and ambiguous as soon as we go into detail: It is clear that corners of the teeth correspond to each other. But on relatively featureless surfaces like the faces of the teeth or the forehead of the skull, it is not clear which points correspond to each other and which do not. Is the relative position within the shape more important than local surface features? How is the correspondence between two skulls with a different number of teeth defined? Can one point correspond to several other points? In Figure 1.3, the red arrows represent correspondences between two skulls that are obvious and well-defined, while the blue arrows show correspondences that are more uncertain.

There is no definite answer to the question if or how correspondence can be defined in these cases and we have to accept that the concept of correspondence cannot be precisely defined down to the last detail. It can even depend on the application. For instance, for a tooth model, we may be interested in the correspondence between all different types of teeth, whereas in a skull model we consider only correspondence between equivalent teeth, e.g. between the canines of one skull and the canines of the other skull.

How can we build shape models based on correspondence, if we cannot even give a precise definition of correspondence? In all but the most simple examples it is impossible to calculate correspondence perfectly. It has to be approximated with a *registration algorithm*. Based on the way the algorithm tries to calculate the correspondence, we can find a pragmatic way to deal with the ambiguities of correspondence in the next section.

1.1.2 Registration

The problem of establishing correspondence is known as the *registration problem* and is one of the central problems in computer vision. A large number of *registration algorithms* have been proposed to address the problem, but it is far from being solved. Most registration methods are based on the following observation:

Suppose that we have complete dense correspondence between two surfaces Γ_1 and $\Gamma_2 \subset \mathbb{R}^3$. That is, we have a map $\Phi : \Gamma_1 \rightarrow \Gamma_2$ that maps every point $x \in \Gamma_1$ to its corresponding point $\Phi(x) \in \Gamma_2$. If this map is bijective, we can represent Γ_2 as a deformation of Γ_1 with Φ :

$$\Gamma_2 = \{\Phi(x) \mid x \in \Gamma_1\} =: \Phi(\Gamma_1). \quad (1.1)$$

This is also referred to as a “warp” or “morph” of Γ_1 . The registration problem is equivalent to finding this map Φ and most registration algorithms try to approximate Φ by minimizing the distance between $\Phi(\Gamma_1)$ and Γ_2 . If we are able to reduce this distance to zero we have found a registration for the two surfaces, which is a correct registration if, additionally, the points x and $\Phi(x)$ correspond and Φ is bijective.

Unfortunately, for all but the easiest cases, we will not be able to reduce the distance to zero and find a perfect registration result. How well the registration result approximates the true correspondence between the surfaces depends on the algorithm. It depends on the way it measures and minimizes the distance between $\Phi(\Gamma_1)$ and Γ_2 and on the way it tries to enforce that Φ is bijective and actually maps only corresponding points onto each other. Each registration algorithm tries to address these problems in its own way and a single best registration algorithm that works perfectly in every situation does not exist. Because the registration problem is not well-defined and typically no ground truth is available for real data it is virtually impossible to compare two registration results or algorithms objectively.

Therefore, we tried to design our own new registration algorithm that works as well as possible for our given task of constructing 3D shape models for medical applications. When correspondence between two shapes is given by the approximated correspondence field Φ , we can define linear combinations of the shapes based on a morph with this field, as in Figure 1.4. Contrary to the linear combination from Figure 1.2, the linear combination is again a valid skull shape.

Upon closer inspection, we see that this linear combination, which should mark exactly the half-point between the two shapes is more similar to the first than the second shape. This is because it is actually a linear combination between the first shape Γ_1 and the warp of this first shape $\Phi(\Gamma_1)$, which does not exactly coincide with Γ_2 for real-world examples of fields Φ that are approximated by a registration algorithm. On one hand this introduces a bias towards Γ_1 , which is an obvious drawback of this way of computing linear combinations of shapes, on the other hand it also holds several important advantages. For instance, in Figure 1.4, we see that, unlike the first skull, the second skull does not have the full set of teeth. Furthermore, due to low resolution the interior structures of the second skull were impossible to reconstruct correctly from the original CT scan, resulting in the artifacts we can see through the skull’s left eye. In this type of linear combinations, these imperfections are replaced by the correct anatomy found in the first skulls.

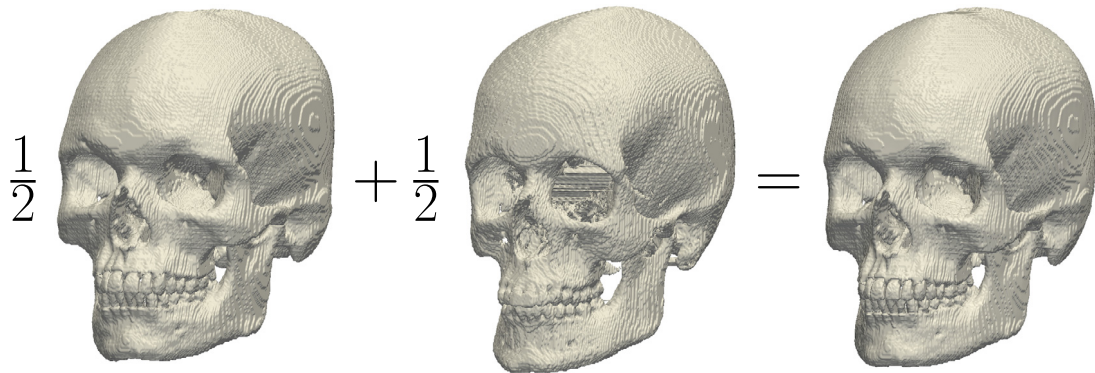


Figure 1.4: Linear combination of two skulls based on correspondence. Contrary to Figure 1.2, the linear combination is again a valid skull.

In the Morphable Model, each shape is represented by a deformation of one common reference shape. The model is based on linear combinations of these deformations. For our skull model, we use the first shape of Figure 1.4 as a reference, because it is the data set with the highest data quality. It was acquired from a very high resolution CT scan, from which the shape was extracted manually with great attention to anatomical detail. Almost all other skull data sets suffer from problems like missing teeth, CT scanning artifacts, or low resolution. Therefore, for this model, it is a great advantage that all shapes are represented by only one anatomically correct reference. Nevertheless, using a single reference introduces a bias towards this reference into the model, and several research groups are working towards at least reducing this bias [71, 7].

The representation of shapes by deformations of a reference also determines the way that we deal with the ambiguities of correspondence mentioned above. Even in places where it is unclear how the correspondence should be defined correctly, we simply use the correspondence between the reference and its deformation by the registration result. In the case of missing teeth, this means that we use the correspondence between the teeth of the reference skull and the deformation of these teeth. Ideally, these should coincide with the teeth that the second skull would have if they were not missing.

How well the deformations of the reference represents the example skulls, both in the defective and in the intact regions, depends on the registration result. This makes the registration algorithm the most important step in building a Morphable Model, and explains the prominent place it took in our research and this thesis. The registration algorithm has to solve the inherently ill-posed problem of establishing correspondence between shapes, which may even contain defects, in a way that allows the construction of a shape model. The better the algorithm performs, the better the resulting model will be. If the shapes are so different that every registration algorithm fails to find meaningful correspondence, they cannot be considered to belong to the same object class and therefore cannot be represented by a common shape model.

1.1.3 Morphable Model

The Morphable Model represents all shapes of an object class by deformation of a common reference shape. These deformations are linear combinations of a set of example deformations, which are usually the registration results of registering the reference to example shapes. The Morphable Model thus describes all shapes by a linear space of deformations. Not all deformations in this linear “model space” represent valid shapes, however, especially if they are far from the example deformations. To model which deformations represent valid shapes, the Morphable Model estimates a normal distribution $\mathcal{N}(\mu, \mathcal{C})$ from the example deformation. This is a normal distribution on the linear model space, but it can be extended to the space of all possible deformations of the reference. Then, all deformations that can be considered plausible according to the normal distribution $\mathcal{N}(\mu, \mathcal{C})$ represent valid shapes of the modeled object class; provided all the examples belong to a class of shapes that are similar enough and approximately normally distributed. If, in addition, the examples represent the object class sufficiently well, the normal distribution $\mathcal{N}(\mu, \mathcal{C})$ represents all members of the object class. This means that given enough representative examples, a Morphable Model aims at representing the class of all possible skulls or all possible femurs etc.

Essentially, the Morphable Model is a statistical model of deformations. Therefore, this model and related models such as the Active Shape Model [23] are referred to as “statistical shape models” or “statistical deformation models”.

1.2 Outline and Contributions

The rest of this thesis is organized according to the chronological order of the steps it takes to build a statistical shape model. We begin by describing in Chapter 2 how the necessary example data sets can be acquired and prepared for registration and model building.

We introduce our registration algorithm in Chapter 3. In line with its importance for building shape models, a large portion of research time and a large portion of this thesis were dedicated to this registration algorithm. The algorithm is based on representing the two shapes that we wish to register by distance functions and registering these distance functions like images. This means that we calculate a vector field which warps one image so that it resembles the other image as closely as possible. This vector field then represents the correspondence of the shapes as well as of the surrounding space. The algorithm is described in depth in Chapter 3. In this context, we are able to present the following contributions:

- We show that when only the distance images are used to represent the surfaces, the registration maps one shape to the other, but often does not map corresponding points onto each other. By including additional feature images like the curvature of the shapes, we can obtain a registration results that maps points in a way that is much closer to our intuitive notion of correspondence [28].
- To ensure that the registration result does not deviate too far from the goal that it should be bijective and optimally even diffeomorphic, every registration algorithm

has to include some kind of regularization. We show that the most straight-forward and widely used regularization term allows unnatural expansion or compression of the shapes. By including a volume preservation term, we obtain a registration result that allows only a very even and consistent change of volume.

- We implement the registration method with a state of the art finite element method, which allows the efficient parallel calculation of the registration result on a locally adaptive grid [28].
- We introduce the registration method as a continuously defined minimization problem, making it independent of the proposed discretization method.
- We show that by introducing an additional statistical regularization term into the registration functional, we can penalize unlikely solutions and make the registration more robust and allow the registration of problematic data sets such as those with missing parts [2].

The statistical regularization term is based on the statistical model and therefore we postpone it until after the introduction of the statistical shape model, which is described in Chapter 4. While we mostly follow the concepts of the Morphable Model [13], a novelty in this thesis is that we introduce the statistical model in the most general way possible: as a statistical model on a general Hilbert space \mathcal{H} . This unifies all possible applications of the same modeling technique into one formulation. This formulation applies directly to continuously or discretely defined surface models of shapes, deformations, image intensity, surface color etc. Choosing any of these models only amounts to choosing the appropriate Hilbert space \mathcal{H} .

Once the model is defined in this general way, we show its application in a variety of scenarios in Chapter 5:

- We show how a surface model can be fitted to a complete or partial surface. For the case that the given surface is only partially defined, we model the flexibility that remains in the model, i.e. we investigate how strongly the given part influences the whole shape [1, 44].
- We fit a model of shape and image intensities directly to CT scans. In order to improve the fitting near the object boundary we include a generic model for the outside of the object [3].
- We include a statistical model into the registration process in form of a regularization term [2].
- For cases in which no prior statistical model is available, we investigate the possibility to build a statistical model directly from problematic and partial data sets [45].
- We show how statistical shape models of bones can be used to design optimal implants for osteosynthesis.

- We show how several statistical models can be linked together, allowing for instance the prediction of faces from skulls by connecting a skull and a face model [57].
- We show how statistical shape models of bones can be used to automatically locate the subchondral bone plate and visualize its density.
- Finally, for the last application, we come back to our introductory example of the broken femur in Figure 1.1, and show how the statistical model can be used to automatically compute an optimal reposition of broken bones from a CT scan.

Many of these applications point towards interesting ideas for future research, which we will discuss in Chapter 6.

1.3 Prior Work

We have included the relevant prior work for each method we use in the respective sections. Here, we would like to mention the works that were most influential or relevant to this thesis. In representing object classes by shape models, the most influential work was the Morphable Model, introduced by Volker Blanz and Thomas Vetter in [13]. Our principles of registration, especially its thorough mathematical treatment was most influenced by the book on image registration by Jan Modersitzki [49]. While the representation of shapes by distance images or, more generally, level set functions can be considered standard, its use in image registration has been pioneered by Nikos Paragios et al. in [55].

The formulation of the registration method as a continuous optimization problem as well as its minimization and discretization with a finite element method is of course influenced by the work of Euler, Lagrange, Ritz, Galerkin, and Hilbert. In their place, we mention the books of Evans [30] and Braess [15]. The specific type of finite element method we use is the local discontinuous Galerkin introduced by Cockburn and Shu in [21].

The idea of incorporating the statistical model itself into the registration process was previously described by Gee and Bajcsy [32] as well as Wang and Staib [73]. An early yet very comprehensive and mathematically thorough introduction to the use and construction of statistical models in medical image analysis is presented by Granander and Miller in [36].

2 Data Acquisition and Preprocessing

The greatest organizational challenge in building a statistical model is acquiring enough data sets to represent the shape variations of the object class we wish to model. As in any other statistical application, the more data sets we can acquire, the better our model will be. Specifying how many data sets are necessary to build a good statistical model depends on the complexity of the shape and the individual data sets. However, a proof of concept for most methods involving statistical shape models can already be achieved with 10 – 20 examples.

2.1 CT Scanning

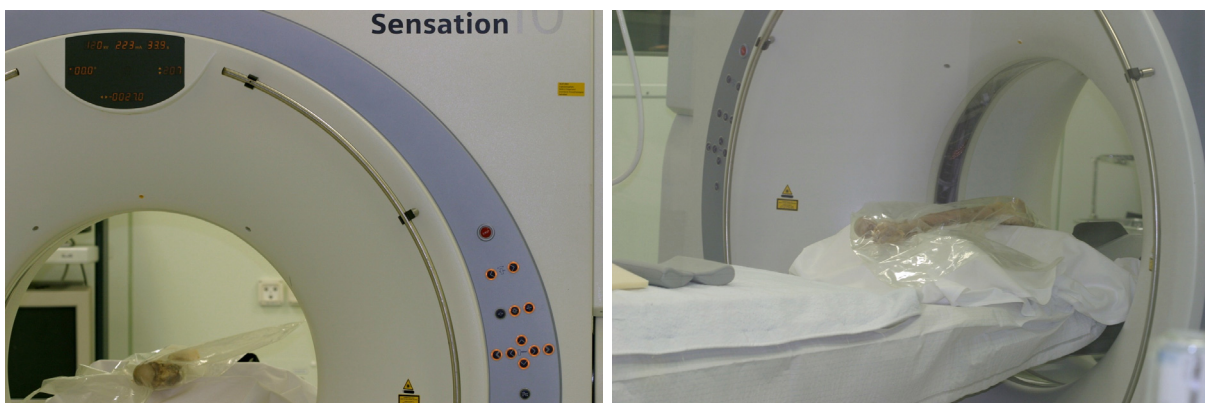


Figure 2.1: CT scanning of isolated femur bones

For our main applications of bone models, the best way to capture the shape and at the same time the density information of bones is with a computed tomography (CT) scanner. It produces a 3 dimensional array of voxels (3D pixels). Each voxel represents a small box in space and the voxel's value represents the ability of the material in this box to block x-rays. The values of the CT image are given in Hounsfield Units (HU) [17]. Air typically has a value of -1000 HU and bone a value of 400 HU or above. Other tissues in the human body have Hounsfield values between around -100 and 100. Therefore, in principle bones can be easily identified in a CT scan based on their high Hounsfield values and any CT scan containing a complete bone can be used as an example data set.

In practice however, things turn out to be more difficult. While thousands of CT scans are performed every day, there is practically never a medical indication to perform a high-resolution CT scan of an entire healthy bone. Clinical CT scans typically contain

broken bones. Additionally, in order to minimize the patient’s exposure to potentially harmful x-rays, the resolution of the scans is typically kept to a minimum.

The amount of radiation necessary to perform a high resolution scan of an entire bone makes it impossible to acquire example scans of healthy bones from living volunteers. Therefore, the only possible source for high quality scans of the entire bones are cadavers. This requires the help of medical partners. We are very thankful for the support from the Anatomical Institute and the University Hospital Basel. The Anatomical Institute provided us with 20 dry (macerated) femur bones (see Figure 2.2) and an additional 25 bones from the medical students’ anatomy courses (see Figure 2.1). These were scanned with a CT scanner at the University Hospital Basel. This provided the valuable data sets without which building the statistical model would not have been possible.



Figure 2.2: Macerated femur bones

To get these bones scanned, we actually had to carry them to the University Hospital ourselves and scan them in a CT scanner that was momentarily unused. The other data sets that we use in this thesis stem from other projects and we received the CT scans directly in digital form, without the need to actually handle human bones.

2.2 Segmentation

While some research groups try to develop shape models directly from medical images [36], we take the more common approach to first separate the object from the background. We are particularly interested in the shape of the object, in other words its outline or surface. By segmenting the image into object and background, we obtain a direct representation of this surface. Secondly, in most cases the background contains superfluous information that we do not wish to model. In our bone example data, the background contains the stretcher of the CT scanner, plastic bags used to wrap the bones, Styrofoam blocks on which the bones were placed, residual soft tissue that is still attached to some bones etc. None of this information is useful for the bone model and could even affect it adversely. Therefore, it should be excluded.

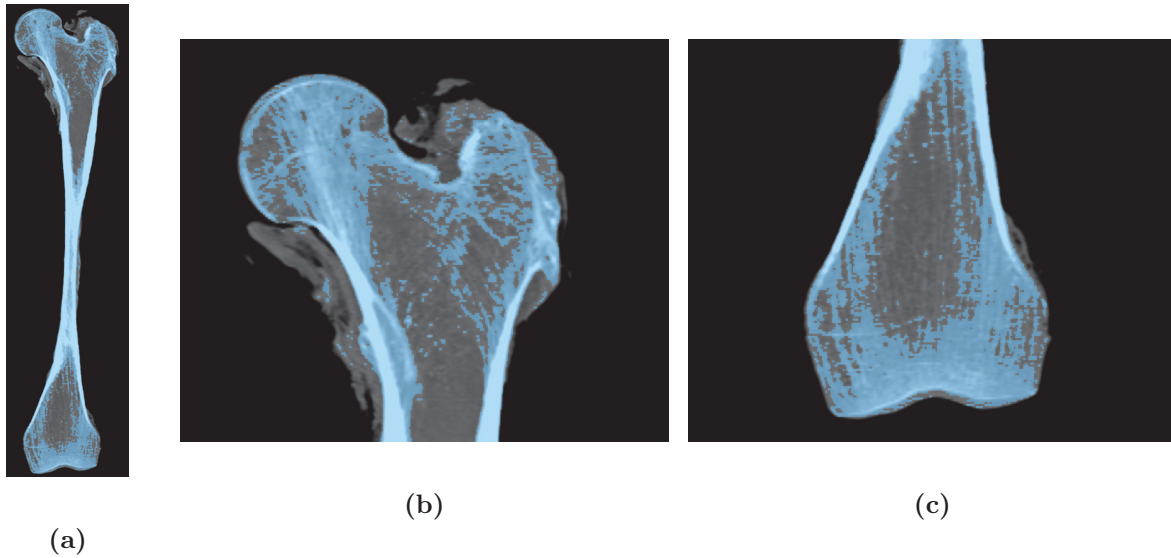


Figure 2.3: Threshold segmentation: At first glance, marking all pixels with a higher intensity than 130 HU identifies the bone in a slice of a CT image (a). Upon closer inspection, we see that many parts of the bone are missed, while some soft tissue is misclassified as bone.

Because bone has a higher density and therefore higher Hounsfield values than all other types of tissue in the body as well as the background objects in the CT scans, the bone can in principle be identified easily by selecting only voxels with Hounsfield value greater than a given threshold. This works reasonably well to get an overall impression of the shape of the object, see Figure 2.3a, where, in a slice of a CT scan, the femur bone is marked in blue by selecting all pixels with an intensity higher than 130 Hounsfield Units. However, upon closer inspection, in Figures 2.3b and 2.3c, we notice that not all of the actual bone tissue is marked as bone. The Hounsfield value of a voxel represents the average density over that voxel. Therefore, a voxel that contains part air and part bone has a much lower value than the theoretic Hounsfield value of bone. This is essentially an aliasing artifact and is known as the “partial volume effect” in the medical community. Therefore, when we choose a threshold value that should, in theory separate bone from other tissue and air, we miss the voxels that are only partially filled by bone. As we can see in Figure 2.3c, this concerns both the inside as well as the outer boundary of the bone. At the same time, we can observe in Figure 2.3b that with this same threshold some of the remaining soft tissue is misclassified as bone. Choosing a lower threshold would misclassify even more soft tissue or even background objects as bone, whereas a higher threshold would miss even more of the actual bone, especially at the boundary in Figure 2.3c. To make this problem even more severe, some of the example bones we received as examples were actually damaged in places where the bone is very thin.

There are many attempts at designing automatic segmentation algorithms to tackle these kinds of problems, and many are implemented in standard software packages such as ITK [39] or Slicer [58]. However, none of these available algorithms, not even the more sophisticated level set methods, were able to completely solve the fundamental problem of

either classifying too much or too little of the image as bone. For many medical applications, this does not pose a significant problem, because the holes in the bone or the additional soft tissue can simply be ignored by a physician. For building a shape model, however, which is supposed to model the complete bone, without holes or residual soft tissue, a more accurate segmentation is necessary. One of the goals of statistical models is to design better segmentation algorithms for the future, but in order to build a model in the first place, the examples have to be segmented somehow and the most reliable method for this proved to be manual segmentation.

Starting from a threshold value that is a relatively good compromise between classifying too many or too few voxels as bone, like that in Figure 2.3, the segmentation is inspected slice by slice, filling the holes and removing the remaining soft tissue manually. While the application Slicer [58] allows for rudimentary image manipulation of 2D slices, this is a very tedious task and a great motivation for continued research in automatic image segmentation.

A typical resulting segmentation result can be seen in Figure 2.4. Unfortunately, the slice-by-slice segmentation introduces additional artifacts when the manual editing is not performed consistently over all slices, see the vertical stripes in the 3D reconstruction Figure 2.4c. With further editing and smoothing, some of our examples have been manually segmented to provide an almost idealized representation of the bone like the bones in Figure 2.5 in the next section. But most bones we used were segmented as accurately as the example in Figure 2.4. This proved to be a good compromise between a reasonable segmentation and a justifiable amount of manual user interaction.

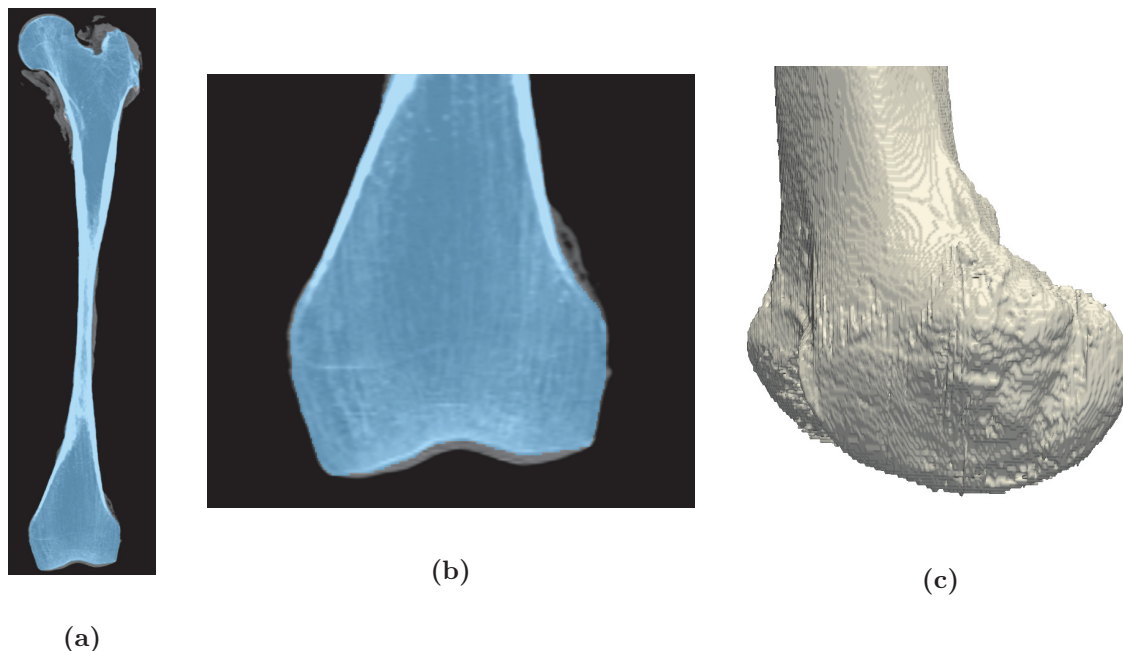


Figure 2.4: Hand segmentation: The bone is correctly identified, but the manual interaction introduces some artifacts, which are visible as vertical stripes in the 3D reconstruction (c).

2.3 Pre-Alignment

Usually, when images or 3D scans are collected for a statistical model, each object is intuitively placed in the middle of the image and in a similar position. However, for accurate registration and model building, the objects should be placed in the same position as accurately as possible. This provides a good initialization for the registration algorithm, and, more importantly, without pre-alignment the statistical model would not only model the shape variations, but also the small differences in position of the examples.

It is hardly possible to perform an exact alignment by hand, for instance by placing the bones in the exact same position in the CT scanner. Therefore, the alignment is performed in the computer, after the CT images have been acquired. While there are a number of automatic alignment algorithms, the most reliable technique is, similar to the segmentation, based on manual user interaction. A representative set of corresponding landmark points is marked on each of the examples. For the femur bone model, we chose the most prominent features described in an anatomy book, see Figure 2.5.

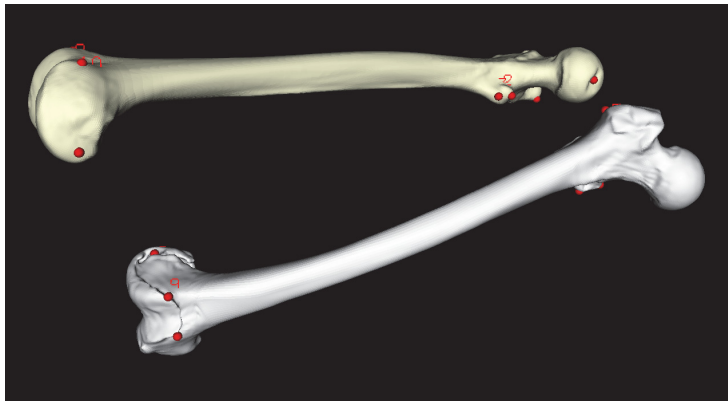


Figure 2.5: Two bones with manually marked landmarks

Once the points are marked on each of the example shapes, we seek a transformation that moves the landmarks and thus the objects they represent as close together as possible. At this point we only want to manipulate the position and not the shape of the objects. Therefore, as possible transformations we admit only rigid motions or similarity transforms, i.e. transformations $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that can be represented as

$$\Phi(x) = \lambda \mathbf{A}x + \mathbf{t}, \quad (2.1)$$

where $\mathbf{A} \in \mathcal{SO}(d)$ is a rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ a translation vector, and $\lambda \in \mathbb{R}^+$ scaling parameter which can take any positive value for a similarity transform and is fixed to $\lambda \equiv 1$ for rigid motions. It depends on the application if scaling should be allowed or not. In cases where the data acquisition can introduce scale inconsistencies, such as in photography or radiography, allowing scaling in the pre-alignment can help remove these inconsistencies. On the other hand, for methods like CT or MR scanning, which produce consistently scaled output, allowing only rigid pre-alignment seems favorable. In any case, the subsequent

task of establishing correspondence between the examples with a registration algorithm is significantly easier if the pre-alignment admits scaling and can thus bring the objects closer together. But, if desired, this scaling can be removed again after registration, resulting in a statistical shape model that preserves the proper scaling of the examples.

In order to align two shapes based on their landmarks, we seek a transformation $\Phi = \lambda \mathbf{A}x + \mathbf{t}$ which minimizes the distance between the transformed landmarks of one shape $\{\Phi(x_1), \dots, \Phi(x_m)\}$ and the landmarks of the other shape $\{y_1, \dots, y_m\}$. This means that we find the parameters of the transform as:

$$(\lambda, \mathbf{A}, \mathbf{t}) = \operatorname{argmin}_{\lambda, \mathbf{A}, \mathbf{t}} \sum_{i=1}^m \|\Phi(x_i) - y_i\|^2 = \operatorname{argmin}_{\lambda, \mathbf{A}, \mathbf{t}} \sum_{i=1}^m \|\lambda \mathbf{A}x_i + \mathbf{t} - y_i\|^2. \quad (2.2)$$

The minimum can be found in a closed form solution, see [72] for details. Traditionally, this form of aligning two shapes is referred to as ‘‘Procrustes Alignment’’, after a character from Greek mythology. Figure 2.6 shows the alignment of the two bones from Figure 2.5. In Figure 2.6a, λ was fixed to 1, resulting in a purely rigid alignment, while in Figure 2.6b, the optimal λ was calculated, resulting in what we call a similarity alignment.

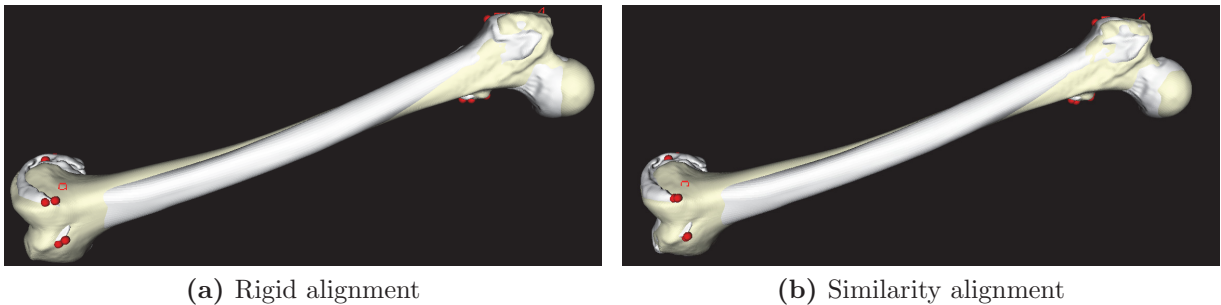


Figure 2.6: Comparison between rigid and similarity alignment: Rigid alignment preserves the original scale of the bones, whereas similarity alignment allows for an isotropic scaling of the aligned bone in order to produce a better match.

In order to align not only two but all the example shapes at the same time, all shapes can be aligned to the common reference. If we wish to make the alignment independent of the chosen landmarks, we can perform an additional alignment after the registration by using all points in the shapes, which are now in correspondence, as landmarks. Additionally, we can make the alignment independent of the reference by performing Generalized Procrustes Analysis [35], in which all shapes are aligned to a mean shape, or rather a mean set of landmarks, which is estimated during the alignment in an iterative process. However, a noticeable improvement in the alignment with this method was only achieved for a model of the skull, for which the landmark selection was rather difficult. For all other object classes we modeled, the regular Procrustes alignment proved to be sufficient.

3 Registration

The most important and most challenging step in constructing shape models from examples is bringing these examples into correspondence. The process of bringing two shapes into correspondence is known as “registration” and consists of finding, for each point on one surface, the corresponding point on the other surface. It may depend on the application what “corresponding” actually means, but typically we assume that corresponding points share a comparable position and function within their respective data sets.

In principle, we wish to establish correspondence between all example objects. The most straight-forward way to accomplish this is to register each example individually to a common reference. The registration algorithm then only needs to establish correspondence between two objects at a time. This is achieved by deforming the reference so that it resembles the other object as closely as possible. The statistical shape model can be built directly from these deformations of the reference shape.

In order to move away from relying on a single reference shape, there are recent developments in group-wise registration and statistical models that use pairwise registration of all examples, see [71, 7] for instance. But as these come at a much greater computational cost, we chose to use the traditional method of registering each example to a common reference.

In most applications, such as in our bone examples, the objects that need to be brought into correspondence are organs that were captured with a medical imaging device, such as a CT or MRI scanner. We propose a method to bring the objects into correspondence by registering a number of feature images derived from these medical images.

As our ultimate goal is the construction of statistical shape models, we are mostly interested in the shape of the organ’s surface, which we represent by the two most prominent feature images in our method: a distance and a curvature image of the surface. Together, they provide a good description of the shape of an object. Other possible feature images, which are then simultaneously registered, can encode additional information about the organs like the original CT or MRI data. Registering all feature images together takes all of this information about the shapes into account.

It is possible to register only one pair of feature images, but our experiments have shown that registering only distance images does not match the details of the surfaces well enough, requiring at least the use of an additional curvature feature image. Registering the original CT or MRI images directly without additional feature images also produces unsatisfying results as a good registration of the image intensity values often does not guarantee a good matching of the object surfaces. There are too many points in the images that do not correspond to each other but share the same intensity value. This problem is less severe if we use several feature images, as there are less non-corresponding points that share the same intensity in *all* feature images. For instance, in the CT and MRI images, all points on the surface and the inside of a bone share a similar intensity, whereas in the distance

image only points *on* the surface share the same intensity. And if we include the curvature image, only points that are on the surface *and* share a similar curvature are considered in the search for corresponding points.

Nevertheless, the registration problem remains ill-posed and we will need to include some type of regularization into our registration method in order to rule out erratic registration results. Using regularization is the standard strategy for trying to solve ill-posed problems. It can be interpreted as including prior knowledge about the possible solutions. The solutions of the registration problem is given in form of a vector field, referred to as the correspondence or deformation field. In the most basic form of our proposed registration method we simply enforce the smoothness of this vector field by controlling the norm of its first derivative. This type of regularization corresponds to the prior knowledge that a good registration result should be smooth. However, it turns out that this regularizer, which is also found in the Demons or Diffusion registration algorithms [67, 49], allows large and unnatural looking volume change. By penalizing volume change we impose our additional prior knowledge that the registration result should not compress or expand the objects excessively. This is achieved by penalizing the linearized volume change caused by the vector field.

One of the main contributions of this thesis is the formulation, discretization, and optimization of the registration problem as a continuous functional, integrating all the different terms described above into a single continuous minimization problem for the deformation field. This formulation allows for the simple enhancement of the scheme with further terms and for a straightforward discretization. We present a memory-efficient and flexible scheme using adaptive finite elements with the *local discontinuous Galerkin* method. This specific type of finite element discretization leads to a very simple formulation even of complex regularization terms and is especially well-suited for speeding up the registration by using non-conforming locally adapted grids and straight-forward parallelization. It is the preferred discretization method of our collaborators at the institute of applied mathematics in Freiburg and is based on software libraries they developed [26, 10, 9, 27].

After introducing the statistical shape model in the next chapter, we will show in Chapter 5 how the model can itself be used to include prior knowledge that is specific to the modeled class of objects into the registration algorithm. In this context the advantages in efficiency and memory consumption of our finite element discretization enable us to perform registrations with large high resolution 3D statistical deformation models that where previously not possible.

However, we have taken great care to separate the registration method from the discretization method we use here. The registration method and the statistical model we will introduce in the following chapter are formulated in the most general way possible and can be used with virtually any discretization method.

3.0.1 Prior Work

Non-rigid registration is a well researched problem. For an overview of registration methods we refer to the survey papers by Zitova and Flusser [77] (image registration), Audette et al. [6] (surface registration), and in particular the book by Modersitzki [49] for a thorough

discussion of variational methods for image registration. The most basic form of our method, i.e. leaving out all the optional terms we will introduce, is closely related to Thirion’s Demons algorithm [67] and Modersitzki’s Diffusion registration algorithm [49].

The idea of surface registration using a distance or level-set representation of surfaces has been introduced by Paragios et al. [55] and the inclusion of additional feature images, especially for parametrized surfaces is used for instance in [43]. The use of curvature images has also been presented in our paper [28].

Volume preserving image registration was introduced by Rohlfing et al. in [60] and Haber and Modersitzki in [37]. Rohlfing et al. include a term penalizing volume change in a B-spline based registration framework, while Haber and Modersitzki enforce strict nonlinear volume preservation in a variational formulation. In our approach, we wish to allow a limited amount of volume change and therefore use a soft constraint, i.e. an additive penalty term. For efficiency, we penalize only the linear part of the volume change and we show that this is equivalent to the linear elastic regularization term first introduced by Broit and Christensen et al. in [16, 20], even though our motivation for using this regularizer does not stem from modelling the organs as elastic bodies.

The use of finite elements for image registration goes back at least as far as [33], and we published a first finite element registration algorithm in [28]. The final model derived in this paper results in an elliptic problem with a non-linear forcing term. The finite element discretization for general elliptic problems has now been employed for decades and can be considered standard. A summary of the standard approach of conforming, continuous finite elements can be found in [15]. We employ a discontinuous finite element approach, which allows us to use non-conforming locally adapted grids with distributed memory parallelization. An overview of this class of schemes can be found in [5]. The method we use is based on the *local discontinuous Galerkin* scheme introduced in [21].

3.1 Registration Method

In this section, we provide a detailed description of our registration method.

3.1.1 Level Set Representation

At its core, the registration method we present is an image registration method and as such can be used directly on images. The first naive approach to bring the example data sets into correspondence is therefore to use this or another image registration method on the original CT images. In this way, the complete images, including the bone surface, in which we are mostly interested, should be automatically brought into correspondence. However, because of the influence of noise, background objects, and the fact that many non-corresponding points share the same intensity value, this approach failed to produce a reliable registration of the bones surfaces in our initial experiments. In addition, for some of the data sets we acquired, we do not have access to the original CT images but only to a surface representation.

For these reasons, we decided to develop a method that is able to directly register two surfaces $\Gamma_0, \Gamma_1 \subset \mathbb{R}^d$. These surfaces can be segmented from medical images as described in Section 2.2 or acquired otherwise. At this point we assume that the surfaces are already rigidly pre-aligned as described in Section 2.3, so that our algorithm only needs to recover the non-rigid component of the registration. Moreover, the pre-registration enables us to choose a common rectangular domain $\Omega \subset \mathbb{R}^d$ which contains all example surfaces. On this domain we can represent each surface Γ by its signed distance function $I : \Omega \rightarrow \mathbb{R}$:

$$I(x) := \begin{cases} \text{dist}(x, \Gamma) & x \in \text{outside}(\Gamma) \\ 0 & x \in \Gamma \\ -\text{dist}(x, \Gamma) & x \in \text{inside}(\Gamma), \end{cases} \quad (3.1)$$

where $\text{dist}(x, \Gamma)$ is the Euclidean distance from x to Γ . When registering open surfaces, for which inside and outside cannot be defined, an unsigned distance function can be used.

Such distance functions can be interpreted as images and registered with an image registration method. In effect, we are back to registering images, but, contrary to the original CT images, the distance images contain only information about the surface and no noise or background information, which made the registration of the original images difficult. This concentration on the information that is most relevant to our problem comes at a price, however: It is only possible if we know the surface Γ . Typically it has to be extracted from the original images by segmentation. Essentially, we are making the registration problem less ill-posed by first addressing the segmentation problem. As it is possible to segment our medical images with modest effort, see Section 2.2, this divide-and-conquer strategy proves to be a good way to reduce the complexity of the registration problem considerably by focusing on the most relevant information.

The aim of the registration algorithm is now to find a vector field Φ such that the target surface's distance function I_1 warped with this deformation field, i.e. $I_1(\Phi(x))$, is as close as possible to the distance function of the reference surface given by I_0 . The registration result of the distance functions then implies a registration of the surfaces they represent. As is customary in most registration algorithms, we represent the vector field as $\Phi(x) = x + u(x)$ and formulate our registration method in terms of the *deformation field* $u : \Omega \rightarrow \mathbb{R}^d$. This is equivalent to finding Φ but has the advantage that the deformation that leaves the object unchanged is represented by $u \equiv 0$.

We formulate the registration problem as a minimization problem. It is shown in [49] that virtually all registration methods can be interpreted in this way. The deformation field u is sought as the minimum of a functional which is the sum of two terms: a *distance* and a *regularization* term. Thus, the registration problem consists of finding the minimum of the functional

$$\mathcal{J}[u] = \mathcal{D}[u] + \mathcal{R}[u], \quad (3.2)$$

with distance term \mathcal{D} and regularization term \mathcal{R} . The distance term measures the distance between the reference and the registration target. At its minimum, the warp of the target is as close as possible to the reference image. The regularization term measures the smoothness or regularity of the registration result u . The smaller it is, the more

regular the solution will be. By minimizing both terms simultaneously, we try to bring the reference and target as close together as possible while keeping the deformation field reasonably smooth. We believe that there is no single generic distance or regularization term that guarantees a good registration in every scenario. The notion of correspondence is application-specific, and the more knowledge about the registration task at hand we can include into the method, the higher the chances will be to obtain a result that meets our requirements.

3.1.2 Function Spaces

We represent the images or distance functions we wish to register as functions $I : \Omega \rightarrow \mathbb{R}$ and the registration results as deformation fields $u : \Omega \rightarrow \mathbb{R}^d$. In practice, these will always be given as discrete images or functions. However, because we formulate our registration method continuously, we should specify what function spaces we assume these continuous functions to belong to. Certainly, if we wish to use the L^2 distance as a distance measure for our functions, we have to assume at least that $I \in L^2(\Omega)$. If we wish to calculate the curvature of the image I , we must furthermore assume that the second derivatives of I exist, at least in the weak sense and therefore have to assume $I \in H^2(\Omega)$.

Similarly, we shall assume that the deformation fields u are at least in the space of square integrable functions $L^2(\Omega, \mathbb{R}^d)$, but as the regularization terms we introduce in the following the derivatives of u , we shall assume that u has at least weak first derivatives, i.e. for the following we assume $u \in H^1(\Omega, \mathbb{R}^d)$. In fact, we shall see in Section 3.2.1 that, with appropriate boundary conditions, a minimum in $H^1(\Omega, \mathbb{R}^d)$ exists for our registration functional.

For more information on function spaces of weak derivatives see, for instance, [4].

3.1.3 Distance Term

The basis for the distance term \mathcal{D} is the L^2 difference between the warp of the signed distance images I_1 and the reference I_0 of the two surfaces to be registered:

$$\|I_1(x + u(x)) - I_0(x)\|_{L^2(\Omega)}^2 := \int_{\Omega} (I_1(x + u(x)) - I_0(x))^2 dx. \quad (3.3)$$

The distance images of two similar surfaces have a similar range of values, especially close to the surfaces, which makes the L^2 distance measure an appropriate choice for their comparison. In order to prevent undesirable effects at the boundary, where the distance function of each surface may be cut off at different values, we bound the distance images at a certain distance b from the surface:

$$I^b(x) := \begin{cases} I(x) & \text{if } I(x) \leq b \\ b & \text{if } I(x) > b, \end{cases} \quad (3.4)$$

and register these bounded distance images instead of the original $I(x)$. The bound $b \in \mathbb{R}$ should be chosen so that the b level set of each surface we want to register is completely

contained inside our computation domain Ω . In this way, each image has the value b on the boundary $\partial\Omega$. For illustration, Figure 3.1 shows such a cut-off distance function for a 2D hand shape.

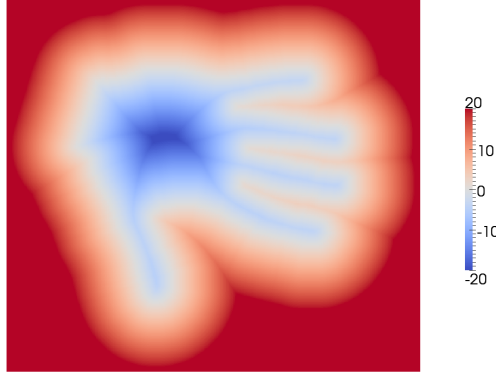


Figure 3.1: A distance image to the outline of a hand, cut off at the value $b = 20$.

3.1.4 Robust Distance Measures

For noisy or otherwise difficult feature images it can be advantageous to use a robust distance measure, which dampens the influence of the overly large differences between the images, see [12] for a review of different robust cost functions. We propose using a robust distance measure based on the Geman-McClure estimator [12, 34], which has been successfully used for medical image registration in [54]. It can be easily realized by weighting the distance measure (3.3) with a term $Q_I(x)$:

$$\mathcal{D}_I[u] := \frac{1}{2} \int_{\Omega} \frac{1}{Q_I(x)} (I_1(x + u(x)) - I_0(x))^2 dx. \quad (3.5)$$

For the Geman-McClure distance measure we have $Q_I(x) = C^2 + (I_1(x + u(x)) - I_0(x))^2$, with a regularization parameter $C \in \mathbb{R}$ which controls the robustness of the measure. A similar term is used in Thirion's Demons algorithm [67], where the norm of the gradient of the image replaces C : $Q_I(x) = |\nabla I_1(x + u(x))|^2 + (I_1(x + u(x)) - I_0(x))^2$. In our experiments, both weights yielded similar results. We have found that for distance images of surfaces that are free from artifacts or excessive noise, it is not necessary to use a robust distance measure. But it proved to be of good use for the additional feature images introduced in the following sections, such as the curvature images.

3.1.5 Curvature Guided Registration

When registering surfaces by means of their distance images, the problem arises that, by definition, the value of the distance function is zero on the whole surface and contains no additional information on the surface. Therefore, the distance function \mathcal{D} is minimized whenever a point on one surface is registered onto a point on the other surface even if the

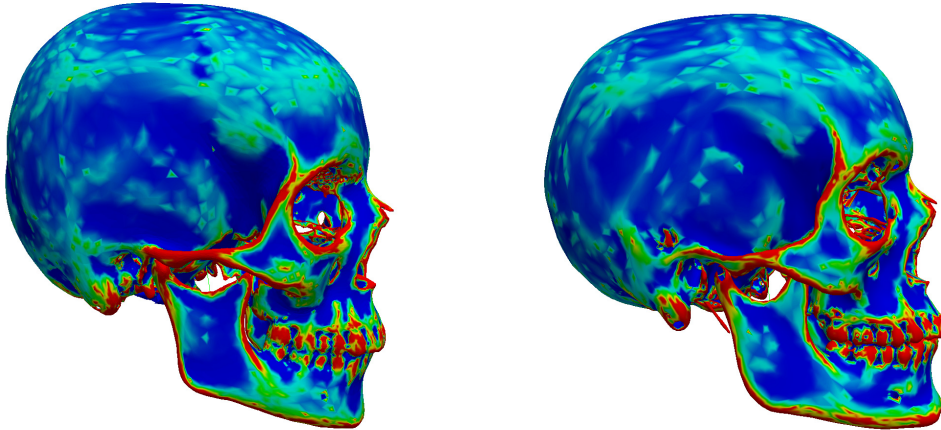


Figure 3.2: Two skulls colored according to their mean curvature. We see that corresponding points have similar mean curvature.

functions or positions of these points do not correspond. In fact, when we try to minimize the registration functional with a gradient descent scheme, the corresponding point is only sought in the direction of the gradient of the distance image, i.e. perpendicular to the surface. This effect is somewhat alleviated by the regularization term, but this is often not enough to obtain a sensible registration. See Figure 3.4 for an example.

For registration of human bones, we wish to establish correspondence between points that have a similar anatomical function. So similar bumps, crests, ridges, etc. should be matched. Such features are well described by the curvature of the surface. In fact, for a large class of objects, corresponding points on two surfaces have similar curvature. Figure 3.2 illustrates this for the mean curvature of human skulls. We use the mean curvature as an additional feature to be matched in our registration algorithm.

With the surfaces represented by their distance images, the curvature is easily calculated by $H(x) = \operatorname{div} \frac{\nabla I}{\|\nabla I\|}$. For each $x \in \Omega$, $H(x)$ is the mean curvature of the level surface passing through x . If x is on the zero level set of I , $H(x)$ is the mean curvature of the surface at that point. Since for distance images $\|\nabla I\| = 1$ almost everywhere, the curvature image H is even more easily computed as $H = \Delta I$, which can be interpreted in the weak sense for $I \in H^2(\Omega)$. If we compute the curvature image as the curvature of the cut off distance functions introduced above, we additionally know that $H \equiv 0$ on the boundary $\partial\Omega$ of our image domain. Figure 3.3 shows the curvature image for the distance function from Figure 3.1.

In differential geometry, the curvature of a surface is actually defined by the curvature tensor. The mean curvature is one of many possible ways to compute a scalar value from this tensor, namely as the mean of its principal components, the principal curvatures. Other possible choices include the Gaussian curvature (the product of the principal curvatures), the shape index, and the curvedness, see [69]. We have chosen the mean curvature because it can be calculated efficiently for distance functions, but obviously other curvature measures can be used instead or in addition to the mean curvature. For instance, in some cases it

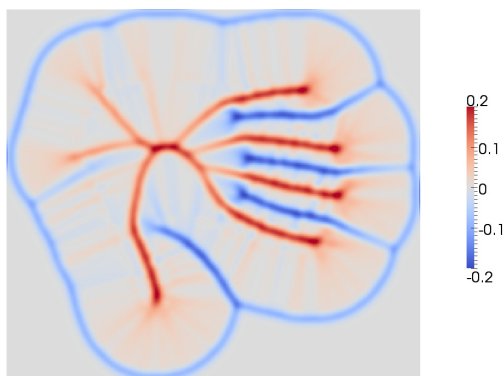


Figure 3.3: The curvature image for the hand distance image from Figure 3.1.

may be of advantage to use a scale-invariant curvature measure as in [69]. On the other hand, we have found that even though the mean curvature is not scale-invariant, when two shapes of different size are registered, the curvature images still encourage a matching of corresponding surface features as any other matching would incur a higher cost in the distance function.

The curvature images are included in the registration process with a distance term analogous to that in Equation (3.5):

$$\mathcal{D}_H[u] := \frac{1}{2} \int_{\Omega} \frac{1}{Q_H(x)} (H_1(x + u(x)) - H_0(x))^2 dx. \quad (3.6)$$

The overall distance measure is then given as $\alpha \mathcal{D}_I[u] + \beta \mathcal{D}_H[u]$ with $\alpha, \beta \in \mathbb{R}^+$ controlling the balance and influence of the distance and curvature images. Figure 3.4 shows a toy example exhibiting the advantage of using the additional curvature image.

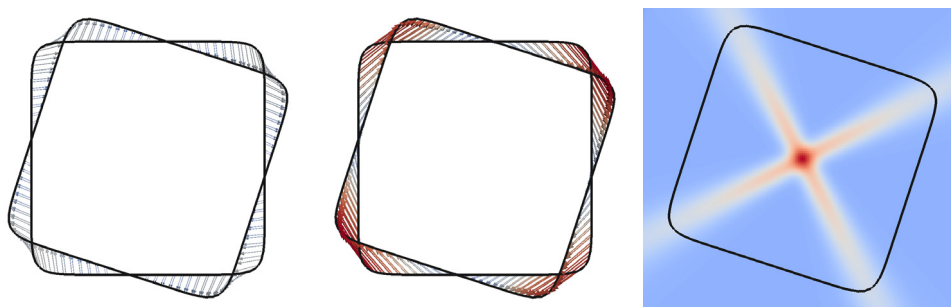


Figure 3.4: Toy example for curvature guided registration: On the left: Without curvature information, the registration of two squares calculates a correspondence field that does not match the corners of the squares. When the mean curvature images as the image on the right are used as additional feature images in the registration, it calculates correspondence between the corners of the two squares, which corresponds much better to our intuitive notion of correspondence.

Additional Feature Images

In an obvious fashion, any number of additional feature images can be added. If we denote the k -th pair of feature images by X_0^k, X_1^k , the full distance term for our registration method is given as:

$$\mathcal{D}[u] := \sum_{k=1}^n \alpha_k \mathcal{D}_{X^k}[u] = \sum_{k=1}^n \frac{\alpha_k}{2} \int_{\Omega} \frac{1}{Q_{X^k}} (X_1^k(x + u(x)) - X_0^k(x))^2 dx, \quad (3.7)$$

with weighting parameters α_k . In our experiments, we have included the original CT scans from which the bone surfaces were segmented as additional feature images where they were available. 2D projections of two such CT scans can be seen in Figure 3.5. Additional image modalities or manual annotations such as images derived from landmarks could also be included, provided they are available for both surfaces to be registered. For multi-modal image pairs, more sophisticated multi-modal distance measures can be employed instead of the L^2 distance measures used in Equation (3.7), cf. [49, 29]. In some cases, it may be necessary to mask out a region of interest so that other objects present in the image do not influence the registration, especially at the boundary $\partial\Omega$.

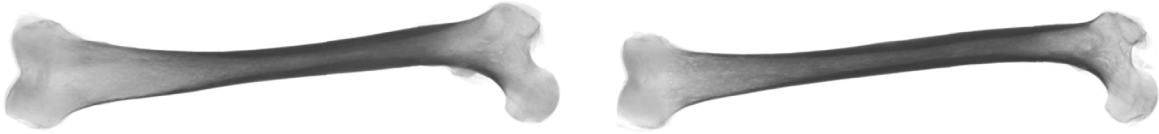


Figure 3.5: 2D projections of CT scans of two femurs.

3.1.6 Regularization Term

Registration is an ill-posed problem and any algorithm trying to minimize a distance measure without enforcing some kind of smoothness or regularity on the solution is bound to fail. We begin by introducing a very basic regularization term, which is later enhanced by adding further terms.

One of the most basic ways to control the smoothness of the deformation field u is through its first derivative Du , which we will, as is customary, denote like a gradient by ∇u . But the reader should bear in mind that $u : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}^d$, and hence $\nabla u(x) \in \mathbb{R}^{d \times d}$. We define the basic regularization term as:

$$\mathcal{R}_g[u] := \frac{1}{2} \int_{\Omega} \|\nabla u(x)\|^2 dx = \frac{1}{2} \sum_{l=1}^d \int_{\Omega} |\nabla u_l(x)|^2 dx, \quad (3.8)$$

where $\nabla u_l(x) \in \mathbb{R}^3$ is the gradient of the l -th coordinate function of u . The smaller $\mathcal{R}_g[u]$ is, the smoother the deformation field u is.

Volume Preservation

While the regularization term (3.8) forces the deformation field to be smooth, it still allows for some quite unnatural deformations. In particular, it allows excessive expansion or compression of the registered object by the deformation field, see Section 3.4.3 for an example. The compression or expansion of the warp $x + u(x)$ can be measured by the determinant of its first derivative $\det[D(x + u(x))]$. A volume preserving deformation field must satisfy $\det[D(x + u(x))] \equiv 1$. Naturally, as we are mostly interested in registering bones of different individuals, which in general do not have the same volume, we do not wish to enforce a strict incompressibility constraint as in fluid dynamics or other registration approaches [37]. Instead, we add a soft incompressibility constraint to our existing regularization term based on the linearization of $\det[D(x + u(x))]$. Thus, volume change is limited but not completely prohibited.

Since $\det[D(x + u(x))] = 1 + \operatorname{div} u + (\text{non-linear terms})$, it follows that the smaller the divergence of u , the closer the linearization of the determinant is to 1 and therefore the field to being volume preserving; provided the values of Du are not too large and the linearization is justified. Therefore we add the square of the deformation field's divergence to the functional:

$$\mathcal{R}[u] := \mu \mathcal{R}_g[u] + \nu \mathcal{R}_d[u] = \mu \frac{1}{2} \int_{\Omega} \|\nabla u\|^2 dx + \nu \frac{1}{2} \int_{\Omega} (\operatorname{div} u)^2 dx.$$

In Section 3.2.4, we will see that the Euler-Lagrange equations for the functional (3.9) correspond to those of the well known linear elastic registration methods. Our introduction of the volume preservation term provides an alternative view on elastic registration. Typically, elastic registration is motivated by arguing that the registered objects can be modeled as elastic objects. We, on the other hand, have found that for inter-subject registration, i.e. the registration anatomic structures of different individuals, it is advantageous to penalize large volume change, even if none of the involved objects or the mapping between them can be considered elastic. Using our proposed volume regularization term results in a more even registration result without over-regularizing or preventing volume change altogether, and when combined with the standard regularization term (3.8) happens to coincide with elastic regularization.

An example of how the penalization of volume change enhances the registration result visibly can be seen in Section 3.4.3. Figure 3.6 shows a toy example of how the volume preservation term works. In this example, we register two ellipses. Without the volume preservation term ($\mu = 0.5, \nu = 0$), one of the ellipses is deformed to match the other ellipse. Obviously, this causes local volume change. When the deformation term is used however ($\nu = \mu = 0.5$), the ellipse is rotated instead of deformed, resulting in an equally good match but without any volume change. Obviously, this is a constructed toy example and the effect for real examples is more subtle.

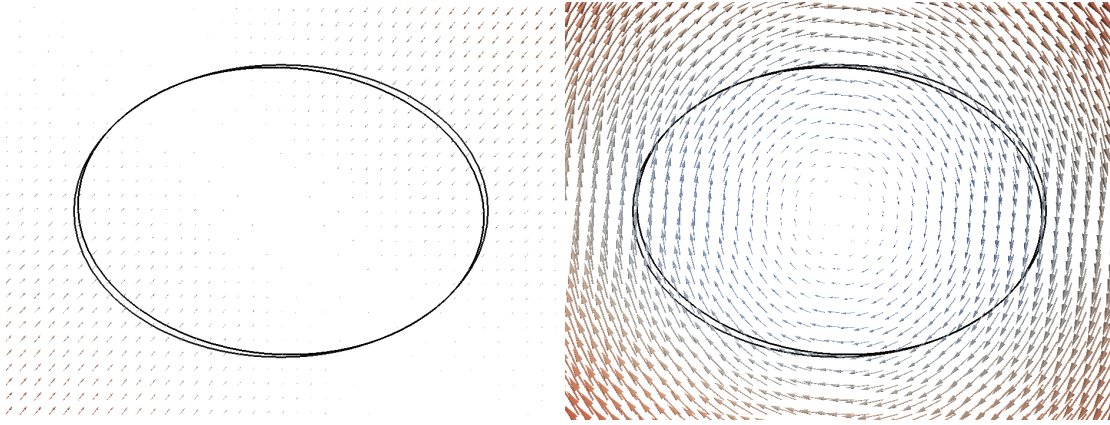


Figure 3.6: Toy example for the volume preservation: Without volume preservation the two ellipses are registered by deformation. With volume preservation, they are rotated, as this causes no volume change at all.

3.1.7 Full Registration Functional

While many more terms may be introduced into the registration functional, and we will in fact introduce an additional statistical regularizer in Chapter 5, let us collect all of the terms we have introduced so far. The full registration functional made up of all the terms introduced above is given as:

$$\mathcal{J}[u] = \mathcal{D}[u] + \mathcal{R}[u] \quad (3.9)$$

$$= \sum_{k=1}^n \frac{\alpha_k}{2} \int_{\Omega} \frac{1}{Q_{x^k}} (X_1^k(x + u(x)) - X_0^k(x))^2 dx \quad (3.10)$$

$$+ \mu \frac{1}{2} \int_{\Omega} \|\nabla u\|^2 dx + \nu \frac{1}{2} \int_{\Omega} (\operatorname{div} u)^2 dx. \quad (3.11)$$

3.2 Minimization Strategy

The optimal registration result that we can achieve with our algorithm is given as the global minimum of the functional $\mathcal{J}[u] = \mathcal{D}[u] + \mathcal{R}[u]$. It is at this minimum that the distance between the deformed reference and the target image is as small as possible, while keeping the irregularity of the deformation field, measured by the regularization term $\mathcal{R}[u]$ as small as possible.

In practice, we try to find the minimum of this functional with an iterative minimization algorithm. If we wish to avoid calculating the derivative of the functional, we may even use an iteration scheme that is based solely on function evaluations, such as those implemented in ITK [39], and just hope that this algorithm finds a suitable registration, starting from an initial guess, typically $u \equiv 0$.

From a mathematical point of view, it is of course interesting to investigate if a minimum of the functional even exists, if it is unique, if it can be further characterized, and how

it can be found most efficiently. While we cannot give a definite answer to all of these questions here, we can at least characterize the problem more precisely.

First of all, we can observe that all terms included in our functional \mathcal{J} are integrals of quadratic terms and hence \mathcal{J} is bounded from below by 0. It follows that the infimum of \mathcal{J} exists and is a nonnegative number $b := \inf \mathcal{J}$, which is bounded above by any value of the functional, say $\mathcal{J}[0]$. The question if a minimum of \mathcal{J} exists reduces to finding out if \mathcal{J} attains its minimum. That is, if there always exists a u such that $\mathcal{J}[u] = \inf \mathcal{J}$.

Secondly, we can easily see that if a minimum of \mathcal{J} exists, it is in general not unique: In a similar fashion to the toy examples in Figures 3.4 and 3.6, we can always construct registration problems that can be easily solved by a simple rotation or translation and that are completely symmetric and therefore admit at least two global minima.

These observations, which require no complicated mathematical theory, hold for virtually any choice of distance measure \mathcal{D} and regularizer \mathcal{R} . We can deduce that even if a minimum of the functional may not exist, we can certainly always use an iterative optimization algorithm, which would at least take us closer to the infimum of the functional.

For our specific choice of distance measure and regularizer, we can, with a little help from the calculus of variations, prove in the following section that at least one minimum of the registration functional exists, if we prescribe boundary conditions. We can also characterize the minimum further by observing that at a minimum u the derivative (i.e. the first variation) of the functional $\mathcal{J}'[u]$ is equal to zero. The equation

$$\mathcal{J}'[u] = 0 \tag{3.12}$$

is known as the *Euler-Lagrange equation* of the functional. The Euler-Lagrange equation of our specific functional will be derived in Section 3.2.3.

3.2.1 Existence of Minimizers of the Registration Functional

In order to prove the existence of a minimizer, we employ a theorem from Evan’s book on partial differential equations [30]. We will paraphrase the theorem to adapt it to our notation, the original can be found in the chapter “Calculus of Variations” in [30].

As with most problems in partial differential equations or the calculus of variation, we need to fix boundary conditions for our problem. The existence theorem in [30] is proved for Dirichlet boundary conditions. That means that the function values are prescribed on the boundary of our domain $\partial\Omega$. As explained in Sections 3.1.1 to 3.1.5 we can assume that all the surfaces we wish to register are contained in a common image domain Ω and that all feature images share the same value on the boundary $\partial\Omega$. Furthermore, the surfaces are all pre-aligned and lie at a reasonably large distance from the boundary. It therefore makes sense to prescribe zero Dirichlet boundary conditions. This means that we prescribe the registration results be zero on the boundary, mapping the boundary onto itself.

The existence theorem proves that a minimizer exists in the appropriate Sobolev function space. We will see below that in our case this is $H^1(\Omega, \mathbb{R}^d)$. As the elements of Sobolev function spaces are in general not defined in a pointwise manner, the boundary conditions have to be understood “in the trace sense”, which means for our case of zero boundary

conditions that the set of admissible functions is that of compactly supported square integrable functions with weak derivative $H_0^1(\Omega, \mathbb{R}^d)$. For more details on Sobolev spaces and traces, see [30] or [4]. Let us state the existence theorem, which is formulated in a very general way:

3.2.1 Theorem (Existence of Minimizer). Let $\mathcal{J}[u]$ be a functional of the form

$$\mathcal{J}[u] = \int_{\Omega} L(\nabla u(x), u(x), x) dx \quad (3.13)$$

for a function $L : \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \bar{\Omega} \rightarrow \mathbb{R}$, called the Lagrangian. If L is convex in the first argument and satisfies the coercivity inequality:

$$L(P, z, x) \geq \alpha \|P\|^q - \beta \quad (P \in \mathbb{R}^{m \times n}, z \in \mathbb{R}^m, x \in \Omega) \quad (3.14)$$

for constants $\alpha > 0, \beta \geq 0$, and if furthermore the admissible set

$$\mathcal{A} := \{u \in H^{1,q}(\Omega, \mathbb{R}^m) \mid u = g \text{ on } \partial\Omega \text{ in the trace sense}\} \quad (3.15)$$

is nonempty, then there exists $u \in \mathcal{A}$ solving:

$$\mathcal{J}[u] = \min_{w \in \mathcal{A}} \mathcal{J}[w]. \quad (3.16)$$

Proof. See [30] □

Applying this theorem proves that there is a minimum for our registration functional:

3.2.2 Corollary. There exists a minimizer $u \in H_0^1(\Omega)$ for the functional

$$\mathcal{J}[u] = \mathcal{D}[u] + \mathcal{R}[u] \quad (3.17)$$

$$= \sum_{k=1}^n \frac{\alpha_k}{2} \int_{\Omega} \frac{1}{Q_{x^k}} (X_1^k(x + u(x)) - X_0^k(x))^2 dx \quad (3.18)$$

$$+ \mu \frac{1}{2} \int_{\Omega} \|\nabla u\|^2 dx + \nu \frac{1}{2} \int_{\Omega} (\operatorname{div} u)^2 dx. \quad (3.19)$$

if $\mu > 0$. That is:

$$\mathcal{J}[u] = \min_{w \in \mathcal{H}_0^1(\Omega, \mathbb{R}^d)} \mathcal{J}[w]. \quad (3.20)$$

Proof. In the notation of Theorem 3.2.1, we have the dimensions $m, n = 3$. The Lagrangian of our registration functional is therefore a function $L : \mathbb{R}^{3 \times 3} \times \mathbb{R}^d \times \bar{\Omega} \rightarrow \mathbb{R}$ defined by:

$$L(P, z, x) = \frac{\mu}{2} \|P\|^2 + \frac{\nu}{2} (\operatorname{trace} P)^2 \quad (3.21)$$

$$+ \sum_{k=1}^n \frac{\alpha_k}{2} \frac{1}{Q_{x^k}} (X_1^k(x + z) - X_0^k(x))^2. \quad (3.22)$$

This Lagrangian is convex in its first argument P . For $\mu > 0$ it furthermore satisfies the coercivity condition from Equation 3.14 with $\alpha = \frac{\mu}{2}, \beta = 0, q = 2$. It follows that the appropriate Sobolev function space is $H^1(\Omega, \mathbb{R}^d) = \bar{H}^{1,2}(\Omega, \mathbb{R}^d)$. With our zero Dirichlet boundary conditions, the admissible set is therefore $\mathcal{A} = H_0^1(\Omega, \mathbb{R}^d)$, which is nonempty. These are all conditions that we need in order to apply Theorem 3.2.1, which proves that $\exists u \in H_0^1(\Omega, \mathbb{R}^d) : \mathcal{J}[u] = \min_{w \in \mathcal{H}_0^1(\Omega, \mathbb{R}^d)} \mathcal{J}[w]$. □

3.2.2 Euler-Lagrange Equations

We know from the previous sections, that our registration functional admits a minimum on $H_0^1(\Omega, \mathbb{R}^d)$ but that the minimum is in general not unique.

For any minimum $u \in H_0^1(\Omega, \mathbb{R}^d)$, we know that:

$$\mathcal{J}[u] \leq \mathcal{J}[w] \quad \forall w \in H_0^1(\Omega, \mathbb{R}^d). \quad (3.23)$$

therefore, for every variation $\varphi \in H_0^1(\Omega, \mathbb{R}^d)$, the function $\varepsilon \mapsto \mathcal{J}[u + \varepsilon\varphi]$ has a minimum at $\varepsilon = 0$. It follows that

$$\mathcal{J}'[u, \varphi] := \left. \frac{\partial}{\partial \varepsilon} \mathcal{J}[u + \varepsilon\varphi] \right|_{\varepsilon=0} = 0 \quad \forall \varphi \in H_0^1(\Omega, \mathbb{R}^d), \quad (3.24)$$

if the derivative exists. The derivative $\mathcal{J}'[u, \varphi]$ is known as the *Gateaux derivative* of \mathcal{J} and Equation (3.24) as the weak form of the Euler-Lagrange equations of our functional \mathcal{J} . By applying the fundamental lemma of calculus of variations [41], we can make this independent of the test function φ and arrive at the strong formulation:

$$\mathcal{J}'[u] = 0. \quad (3.25)$$

This is a system on nonlinear partial differential equations (PDEs) that is satisfied at each critical point of the functional. We will try to solve it in order to find such critical points. As it is very difficult to solve such a nonlinear system of PDEs directly, we employ the method of gradient flow: We introduce an artificial time variable t and try to solve the time-dependent partial differential equation:

$$\partial_t u = -\mathcal{J}'[u] \quad (3.26)$$

for a function $u(x, t)$, given an initial solution $u^0 = u(x, 0)$. This gradient flow will be discretized in time, resulting in an algorithm that is essentially a gradient descent algorithm for minimizing the registration functional \mathcal{J} . The discretized gradient flow can then be seen either as an iterative method to solve the nonlinear Euler-Lagrange equations or as a minimization algorithm for the functional \mathcal{J} . If it does find a minimum, and which minimum it finds if there are several, depends on the initial guess u^0 and other factors such as the time discretization method. For the registration experiments we performed, it always found a global or at least a suitable local minimum with an initial guess of $u^0 \equiv 0$. Furthermore, because we calculate the gradient flow in the direction opposite to \mathcal{J}' , it finds a minimum and no other critical point like a maximum or saddle point of the functional \mathcal{J} , unless $\mathcal{J}'[u^0]$ happens to be zero.

The finite element method we will introduce in Section 3.3.2 is based on the weak formulation of the Euler-Lagrange equations (3.24). We will derive both the strong and weak formulation in the next section by calculating the Gateaux derivatives of all the individual terms of our functional. Calculating the derivatives is necessary for most other types of minimization approaches as well.

3.2.3 Derivatives

Using the standard methods from the calculus of variations we compute the Gateaux derivatives for each of the terms that make up our registration functional \mathcal{J} .

Distance Measure

As the individual terms of the combined distance measure Equation (3.7) are all of the same form, it suffices to calculate the first variation of the distance measure for only one feature image $\mathcal{D}_I[u]$ defined in Equation (3.5).

$$\mathcal{D}'_I[u, \varphi] = \frac{d}{d\varepsilon} \mathcal{D}'_I[u + \varepsilon\varphi] \Big|_{\varepsilon=0} \quad (3.27)$$

$$= \frac{d}{d\varepsilon} \frac{1}{2} \int_{\Omega} \frac{1}{Q_I} (I_1(x + u(x) + \varepsilon\varphi(x)) - I_0(x))^2 dx \Big|_{\varepsilon=0} \quad (3.28)$$

$$= \int_{\Omega} \frac{1}{Q_I} (I_1(x + u(x)) - I_0(x)) \nabla I_1(x + u(x)) \varphi(x) dx. \quad (3.29)$$

The dependency of the weighting term Q_I on u is neglected for reasons of simplicity and efficiency. Differentiating this term, which serves only as a weight of the L^2 distance measure, would make the Euler-Lagrange equations much more complicated without any gain in registration accuracy. The terms of the other feature images are derived analogously, resulting in the following derivative of the distance measure from Equation (3.7):

$$\mathcal{D}'[u, \varphi] = \sum_{k=1}^n \int_{\Omega} \frac{\alpha^k}{Q_{X^k}} (X_1^k(x + u(x)) - X_0^k) \nabla X_1^k(x + u(x)) \varphi(x) dx. \quad (3.30)$$

Regularization Term

We will continue by calculating the first variation of the regularization term $\mathcal{R}_g[u]$ defined in Equation (3.8).

$$\begin{aligned} \mathcal{R}'_g[u, \varphi] &= \frac{d}{d\varepsilon} \mathcal{R}_g[u + \varepsilon\varphi] \Big|_{\varepsilon=0} \\ &= \frac{d}{d\varepsilon} \frac{1}{2} \int_{\Omega} \|\nabla u(x) + \varepsilon \nabla \varphi(x)\|^2 dx \Big|_{\varepsilon=0} \\ &= \int_{\Omega} \nabla u : \nabla \varphi dx := \sum_{i=1}^d \int_{\Omega} \nabla u_i \cdot \nabla \varphi_i dx, \end{aligned} \quad (3.31)$$

The divergence regularization term introduced in Equation (3.9) is differentiated as:

$$\begin{aligned} \mathcal{R}'_d[u, \varphi] &= \frac{d}{d\varepsilon} \frac{1}{2} \int_{\Omega} (\operatorname{div} u + \varepsilon \operatorname{div} \varphi)^2 dx \Big|_{\varepsilon=0} \\ &= \int_{\Omega} \operatorname{div} u \operatorname{div} \varphi dx. \end{aligned} \quad (3.32)$$

Therefore, the derivative of the regularization functional $\mathcal{R}[u]$ from Equation (3.9) is:

$$\begin{aligned}\mathcal{R}'[u, \varphi] &= \mu \mathcal{R}'_g[u, \varphi] + \nu \mathcal{R}'_d[u, \varphi] \\ &= \int_{\Omega} \mu \nabla u : \nabla \varphi + \nu \operatorname{div} u \operatorname{div} \varphi \, dx.\end{aligned}\quad (3.33)$$

3.2.4 Strong Derivative

In order to derive the strong formulation, for the functional, we first apply integration by parts on the derivative of the regularization term Equation (3.33). Assuming zero Neumann or Dirichlet boundary conditions, the boundary terms vanish and we get:

$$\mathcal{R}'[u, \varphi] = - \int_{\Omega} \mu \Delta u \cdot \varphi + \nu \nabla \operatorname{div} u \cdot \varphi \, dx. \quad (3.34)$$

Now we can collect all the terms making up the weak derivative

$$\mathcal{J}'[u, \varphi] = \mathcal{D}'[u, \varphi] + \mathcal{R}'[u, \varphi] \quad (3.35)$$

and apply the Fundamental Lemma of the Calculus of Variations to each component φ_i of φ which leads to the strong form of the Euler-Lagrange equation:

$$\mathcal{J}'[u] = 0 \quad \Leftrightarrow \quad \mathcal{R}'[u] = -\mathcal{D}'[u] \quad \Leftrightarrow \quad (3.36)$$

$$-\mu \Delta u - \nu \nabla \operatorname{div} u = - \sum_{k=1}^n \frac{\alpha_k}{Q_{X^k}(x)} (X_1^k(x + u(x)) - X_0^k(x)) \nabla X_1^k(x + u(x)). \quad (3.37)$$

This is the well known equilibrium equation from linear elasticity [30] with a nonlinear forcing term on the right hand side, which is the derivative of the distance term \mathcal{D} . Hence, this is a second order semilinear partial differential equation. The operator $\mathcal{J}'[u]$ is elliptic for $\mu > 0$, $\nu \geq 0$ making the gradient flow $\partial_t u + \mathcal{J}'[u] = 0$ a parabolic partial differential equation.

3.3 Discretization

We proceed by showing how the gradient flow $\partial_t u = -\mathcal{J}'[u]$, with which we wish to minimize our registration functional can be discretized in order to actually perform a registration on a computer. We will begin by proposing a time discretization, which is independent of the space discretization, before finally introduce our finite element space discretization.

3.3.1 Time Discretization

As is customary with parabolic partial differential equations, we perform time discretization by treating the gradient flow $\partial_t u = -\mathcal{J}'[u]$ like an ordinary differential equation (ODE) in the (artificial) time variable t and applying standard methods for discretization of ODEs.

That is, we approximate the time dependent function $u(x, t)$ by a discrete number of functions $u^m(x)$, which depend only on the spatial variable x . u^m represents the solution at the m -th timestep, and u^0 is the initial solution of our problem. The timestep size $\tau \in \mathbb{R}^+$, i.e. the distance in time between to solutions u^m and u^{m+1} need not necessarily be uniform, but if it is, we have $u^m(x) = u(x, m\tau)$.

The most straight-forward discretization scheme for ODEs, which is known as the *Euler Scheme*, discretizes the time derivative as $\partial_t u \approx \frac{u^{m+1} - u^m}{\tau}$. The time-discretized gradient flow is then given as

$$\frac{u^{m+1} - u^m}{\tau} = -\mathcal{J}'[u^m] \quad \text{or} \quad \frac{u^{m+1} - u^m}{\tau} = -\mathcal{J}'[u^{m+1}], \quad (3.38)$$

depending on whether we evaluate \mathcal{J}' at the old or the new timestep. In an iterative fashion, starting with the initial solution u^0 , each subsequent time step u^{m+1} is calculated from the previous one u^m as:

$$u^{m+1} = u^m - \tau \mathcal{J}'[u^m] \quad \text{or} \quad u^{m+1} = u^m - \tau \mathcal{J}'[u^{m+1}]. \quad (3.39)$$

In the first case, which is known as an *explicit* Euler scheme, the new solution u^{m+1} can be computed directly from the old one, by simply evaluating $\mathcal{J}'[u^m]$. Unfortunately, because the derivative of our regularizer \mathcal{R}' is a second order differential operator, this scheme becomes unstable if the time step $\tau > ch^2$, where c is a constant and h describes the accuracy of the space discretization [15]. For finite element or finite difference grids, h denotes the grid size. This means that for accurate grids, the time step size has to be chosen extremely small, resulting in a very large number of iterations necessary to reach a solution.

The other possibility, the *implicit* Euler scheme is stable for all time step sizes, but it requires the solution of a large nonlinear system of equations, as the new solution u^{m+1} appears not only on the left hand side, but also inside the nonlinear operator $\mathcal{J}'[u^{m+1}]$. Therefore, we propose to use a *semi-implicit* time stepping scheme: We split up the functional $\mathcal{J}[u]$ into an explicit part $\mathcal{J}_{\text{expl}}[u]$, containing all terms with nonlinear derivatives and an implicit part $\mathcal{J}_{\text{impl}}[u]$ containing all terms with linear derivatives. Most importantly, these linear terms include the second order derivatives of the regularizer, which are responsible for the time step restriction in the explicit scheme. The iteration scheme, in its simplest form, is then defined as:

$$u^{m+1} = u^m - \tau \mathcal{J}'_{\text{impl}}[u^{m+1}] - \tau \mathcal{J}'_{\text{expl}}[u^m]. \quad (3.40)$$

In our case this means that $\mathcal{J}_{\text{expl}} = \mathcal{D}$ and $\mathcal{J}_{\text{impl}} = \mathcal{R}$. If additional terms are introduced, they can be added to either $\mathcal{J}_{\text{expl}}$ or $\mathcal{J}_{\text{impl}}$, depending on their linearity. For instance, the statistical regularizer we will introduce later, will be added to $\mathcal{J}_{\text{impl}}$. With this semi-implicit scheme, each iteration step now requires the solution of a *linear* system of equations. This scheme still has a time step restriction, but it is much less severe than that of the purely explicit scheme.

The Euler scheme is a first-order scheme for solving ODEs. Higher order schemes, such as the IMEX Runge-Kutta schemes [56] exist, and we use them in our implementation,

with the same semi-implicit treatment of the operator \mathcal{J}' as for the Euler scheme. Such higher order schemes are able to find a minimum of the registration functional in fewer iterations.

In our experiments, we use the initial guess $u^0 \equiv 0$ and run the iteration scheme until a convergence criterion is met or a predefined number of iterations have been performed.

3.3.2 Finite Element Space Discretization

The semi-implicit time discretization introduced in the previous section discretizes only the (artificial) time variable and essentially provides an iteration scheme to find a minimum for our registration functional \mathcal{J} , thus solving the nonlinear Euler-Lagrange equations. Each semi-implicit iteration step according to Equation (3.40) does however require the solution of a system of linear partial differential equation. That is, in each iteration step, we need to find $u \in H_0^1(\Omega, \mathbb{R}^d)$ such that Equation (3.40) is satisfied on Ω . J'_{impl} is a second order elliptic differential operator. All nonlinear terms are contained in $J'_{\text{expl}}[u^m]$, making this a linear system of PDEs for u^{m+1} .

Nevertheless, we cannot readily solve even this linear system analytically, and will have to discretize it in order to compute an approximate solution on a computer. This discretization is independent of the time discretization and is usually referred to as “space discretization” as it discretizes the space variable $x \in \Omega$. The most straight-forward discretization scheme is most certainly created by approximating the deformation fields $u \in H_0^1(\Omega, \mathbb{R}^d)$ by a discrete vector field defined on a regular grid, for instance on the pixel or voxel grid of the original medical images. The derivatives contained in J'_{impl} can then be discretized by finite differences or even replaced by a smoothing convolution, cf. [67, 49]. This is the type of discretization scheme we first implemented and used for instance in [2].

While these or many other discretization methods, such as methods based on splines or wavelets etc. may very well be used on this problem, we propose a finite-element discretization, because it offers a few important advantages for our specific problem, which will be discussed in the remainder of this Chapter. The basic idea of the finite element method is to approximate the solution space of the problem, in our case $H_0^1(\Omega, \mathbb{R}^d)$ by a finite dimensional subspace $V_h \subset H_0^1(\Omega, \mathbb{R}^d)$ spanned by a set of basis functions $\{\varphi_1, \dots, \varphi_N\}$. While in principle any set of basis functions could be used, the basis functions are usually constructed based on a tessellation, i.e. a grid approximation of the domain Ω . A great advantage of this method over the image-based methods mentioned above is that this grid does not need to be uniform. Instead, it can be chosen to be locally adaptive, which means that the elements of the grid can have a different size in different areas of Ω and therefore the grid can be very fine-meshed in some places and very coarse in other places. This allows us to concentrate the available computation and memory resources on the areas of Ω for which a more accurate solution is required. While we still compute a correspondence field on the whole domain Ω and will make use of it in some of the applications in Chapter 5, we would like to focus our calculation on the vicinity of the actual surfaces of our objects and calculate only a coarse solution further away. This results in considerably lower memory consumption of the discretized functions, as no precious resources are wasted on computing a very accurate solutions in areas of only moderate interest. In fact the statistical

regularization term for our registration functional that we will introduce in Chapter 5 was originally introduced for an image-based uniform discretization in [2], where it could only be applied to 2D images, because of the vast memory requirements of 3D deformation fields with a uniform high resolution. With locally adaptive grids, it becomes possible to store these models of 3D deformation fields in the memory of a standard computer, while retaining a comparable resolution around the surface of interest.

The specific finite element method we use is known as the local discontinuous Galerkin (LDG) method. It differs from traditional finite element methods in that it allows discontinuous basis functions φ_i , which gives us additional freedom in choosing the adaptive grids and allows for an easy splitting of the grid for parallel computations. Furthermore, it turns out that this method allows for a convenient discretization of the elastic regularization term, which will be derived in the next section.

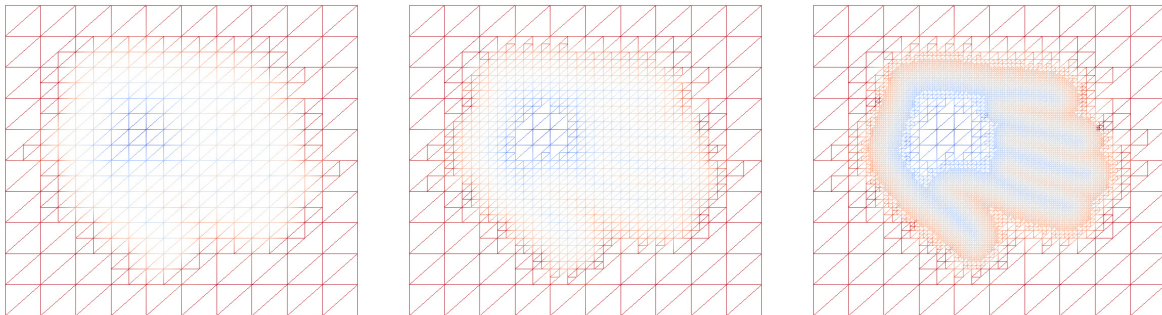


Figure 3.7: Locally adaptive multiresolution

Locally Adaptive Multiresolution Strategy

The local grid adaptation can be embedded into a multiresolution strategy, which is a popular method used in many computer vision applications such as registration or optical flow calculations, see [18] for example. The solution of the problem is calculated on several levels of resolution, starting with a very coarse discretization, on which the solution of the problem can be calculated very quickly, and on which the optimization is less susceptible to getting trapped in local minima. Then, the solution from the coarse resolution is used as the initial guess for a finer discretization. In this way, a minimum of the cost functional is typically found much faster, and because the solution from the previous resolution level provides a good initialization, the chances of finding the desired global minimum instead of some unwanted local minimum are much higher compared to starting the calculation directly on the finest grid with an arbitrary initial guess.

We use this strategy in a locally adaptive way. The first solution is calculated on a coarse uniform grid, starting with the initial guess $u^0 \equiv 0$. All subsequent solutions are calculated on grids that are refined locally. This means that these grids do not have a higher resolution everywhere, as in the traditional multiresolution method, but only in areas where an accurate solution is needed. In our case, we know that we wish to calculate an

accurate solution around the object surface. Therefore, when moving to a higher resolution, we refine the grid only in areas where the absolute value of the distance function I_0 is below a threshold, as these areas are close to the surface. The threshold is decreased after each refinement step, resulting in a mesh whose refinement becomes more and more focused on the surface of interest. A series of such multiresolution grids is shown in Figure 3.7. The refinement is continued until a resolution has been reached that meets the accuracy requirements of the specific registration task.

For other problems, in which it is not clear a-priori which areas of the computation domain should be refined, it is possible to develop local a-posteriori error estimators, which indicate the regions in which the solution is not accurate enough and could be improved by local refinement, see [15] for details.

3.3.3 Local Discontinuous Galerkin Finite Element Discretization

The basic idea of the finite element method is to construct a function space $V_h \subset H_0^1(\Omega, \mathbb{R}^d)$ of finite dimension N and to solve the Euler-Lagrange equation $\mathcal{J}'[u] = 0$ not on $H_0^1(\Omega, \mathbb{R}^d)$ but on its finite-dimensional subspace V_h . We will denote the discrete solution also with u . After choosing basis functions $\varphi_1, \dots, \varphi_N$, a function $u \in V_h$ can be represented as $u(x) = \sum_{k=1}^N u_k \varphi_k(x)$ with a vector of degrees of freedom (DOF) denoted by $\mathbf{u} = (u_k)_{k=1}^N$. Each iteration of our semi-implicit time stepping scheme Equation (3.40) then becomes a linear system of equations for the DOF vector $\mathbf{u} \in \mathbb{R}^N$.

We construct the finite dimensional function space V_h as the space of piecewise polynomial functions based on a tessellation $\mathcal{T} = \{T_i\}_{i \in \mathcal{I}}$ of the image domain Ω . A tessellation of Ω is a collection of geometric primitives T_i which fills Ω with no overlap or gaps, see Figure 3.7 for instance. Typically, these are triangles or rectangles in 2D and tetrahedra or hexahedra in 3D. For a given polynomial degree $q \in \mathbb{N}$, we now define the discrete function space V_h^q as the space of all functions $v : \Omega \rightarrow \mathbb{R}^d$ that are polynomials of degree q on each element T_i of the tessellation. Overall, v is piecewise polynomial. In symbols this can be expressed as:

$$V_h^q := \{v : v|_{T_i} \in [P_q(T_i)]^d \text{ for } i \in \mathcal{I}\}, \quad (3.41)$$

where $P_q(T_i)$ denotes the space of polynomials on the element T_i of order q .

In the standard finite element method, the Galerkin method [15], there is the additional requirement that the functions in V_h^q have to be continuous on Ω , i.e. $V_h^q \subset C^0(\Omega, \mathbb{R}^d)$. This implies additional restrictions on the tessellation \mathcal{T} . Here, we use an extension of the Galerkin method: the *Local Discontinuous Galerkin* (LDG) method, which does not require this continuity assumption, so that there is very little restriction on the tessellation. For instance, the LDG method allows for “hanging nodes” in the tessellation. A hanging node is a node in the tessellation that is not a vertex of all its adjoining elements. Instead, it “hangs” on the edge or face of an element. Figure 3.7 contains several hanging nodes. For instance, there are two hanging nodes in the middle of the rightmost column of triangles of the figure. In the continuous Galerkin method, hanging nodes are not permitted, which makes the use of adaptive rectangular and hexahedral meshes cumbersome. Furthermore, the LDG method allows us to choose orthogonal basis functions $\{\varphi_l\}$ of V_h^q . These are

easily constructed by choosing on each element T a polynomial basis $\{\phi_{T,1}, \dots, \phi_{T,r}\}$ of $[P^q(T)]^d$ satisfying:

$$\int_T \phi_{T,i}(x) \cdot \phi_{T,j}(x) dx = |T| \delta_{ij}, \quad (3.42)$$

where $|T|$ denotes the volume of T and δ_{ij} the Kronecker delta. A basis function φ_l of V_h^q is then chosen to coincide with a polynomial basis function $\phi_{T,i}$ on one element T and vanish on all other elements. As these basis functions are discontinuous, they could not have been used with the standard Galerkin method, even though orthogonal basis functions have many advantages, for instance in the construction of discrete statistical models in Section 4.3.

On the downside, the discontinuous ansatz space of the LDG method requires a discretization of the higher order derivatives of our regularization term \mathcal{R} that is slightly more involved than in the standard Galerkin method. We briefly sketch the LDG approach when applied to a partial differential equation of the form

$$\mathcal{R}'[u] = F \quad (3.43)$$

$$\Leftrightarrow -\mu \Delta u - \nu \nabla \operatorname{div} u = F, \quad (3.44)$$

where F combines all lower order terms. This corresponds to the strong form of the Euler-Lagrange Equation (3.36) of our registration functional \mathcal{J} , and if we replace u by the timestep solution u^{m+1} it also corresponds to the system of equations we have to calculate in each time step of our semi-implicit scheme (3.40). The main step we need to take in order to be able to discretize this form of equation with the LDG method is rewriting the second order partial differential equation as a first order system for the vector-valued functions u, w_1, \dots, w_d . Equation (3.43) is equivalent to the system:

$$w_k - \nabla u_k = 0 \quad (3.45)$$

$$-\sum_{l=1}^d \partial_l (\mu w_{kl} + \nu w_{lk}) = F_k, \quad (3.46)$$

with $k = 1, \dots, d$. In this system, the elastic regularization term \mathcal{R}' is represented by the very compact expression $\partial_l (\mu w_{kl} + \nu w_{lk})$. In the weak formulation provided below, this leads to a formulation involving only first order derivatives on each element $T \in \mathcal{T}_h$. As we do *not* assume zero boundary conditions on each element T , the weak formulation includes boundary terms for the cell boundary ∂T . These boundary terms are handled by numerical fluxes over the cell boundaries in the LDG method. Focusing on a single element T of the tessellation, the corresponding weak form of Equation (3.46) takes the following form if we define the l -th component of the vector-valued function W_k , which represents the

regularization term, as $W_{kl} := (\mu w_{kl} + \nu w_{lk})$.

$$\int_T w_k \cdot \varphi + \int_T u_k \operatorname{div} \varphi - \int_{\partial T} \hat{u}_k \varphi \cdot \mathbf{n} = 0 \quad \forall \varphi \in H_0^1(\Omega, \mathbb{R}^d), \quad (3.47)$$

$$\int_T W_k \cdot \nabla \psi - \int_{\partial T} \widehat{W}_k \cdot \mathbf{n} \psi = \int_T F_k \psi \quad \forall \psi \in H_0^1(\Omega). \quad (3.48)$$

$$\Leftrightarrow: \int_T W : \nabla \varphi - \int_{\partial T} \widehat{W}_n \cdot \varphi = \int_T F \cdot \varphi \quad \forall \varphi \in H_0^1(\Omega, \mathbb{R}^d). \quad (3.49)$$

\mathbf{n} denotes the outer surface normal on ∂T . The fluxes \hat{u}_k and \widehat{W}_k can for example be taken as averages of the values on both sides of the boundary of T or using suitable one-sided values. For more details on the LDG method see [21, 5]. As explained in Section 3.2.1, we use zero Dirichlet conditions boundary conditions for the deformation field u on the domain boundary $\partial\Omega$.

The remainder of the method follows the standard procedure of the Galerkin method. If we choose in Equation (3.47) $u = u^{m+1}$, the weak form of the lower order term F for each semi-implicit iteration step is easily derived from Equations (3.36) and (3.40) as:

$$\int_T F \varphi = - \int_T \frac{1}{\tau} (u^{m+1} - u^m)_k \cdot \varphi - \int_T \mathcal{J}'_{\text{expl}}[u^m]_k \cdot \varphi \quad (3.50)$$

$$= - \int_T \frac{1}{\tau} (u^{m+1}(x) - u^m(x))_k \cdot \varphi(x) dx \quad (3.51)$$

$$- \int_T \sum_{l=1}^n \frac{\alpha_l}{Q_{X^l}} (X_1^l(x + u^m(x)) - X_0^l(x)) \nabla X_1^l(x + u^m(x))_k \cdot \varphi(x) dx. \quad (3.52)$$

Next, we restrict the system to the finite dimensional subspace $V_h^q = \text{span}\{\varphi_1, \dots, \varphi_N\}$, replacing each function $u \in H_0^1(\Omega, \mathbb{R}^d)$ by its approximation $\sum_{i=1}^N u_i \varphi_i \in V_h^q$, and using the basis functions $\{\varphi_1, \dots, \varphi_N\}$ as test functions in Equations 3.47 and 3.49. That is, it is not necessary to solve the equations for all $\varphi \in H_0^1(\Omega, \mathbb{R}^d)$, it suffices if they are satisfied for all basis functions φ_j , $j = 1, \dots, N$. All derivatives in the system are first derivatives of the basis functions φ_j , which can be calculated analytically, because on each element T , the basis functions are polynomials of degree q . Thus, the system of equations 3.47 and 3.49 with the term $\int_T F \cdot \varphi$ defined by Equation (3.50) becomes a linear system of equations for the DOF vectors of the discrete functions u, w_1, \dots, w_d , which can be solved on a computer.

For the sake of completeness, we produce the full discrete system of equations, where the k -th component of a function or vector v_i is denoted by $v_{i,k}$. The fully discretized form of the weak equation system (3.47) – (3.49) is given as:

$$\begin{aligned} \int_T \sum_i w_{k,i} \varphi_i \cdot \varphi_j + \int_T \sum_i u_i^{m+1} \varphi_{i,k} \operatorname{div} \varphi_j - \int_{\partial T} \sum_i \hat{u}_i^{m+1} \varphi_{i,k} \varphi_j \cdot \mathbf{n} &= 0 \quad \forall j \in \{1, \dots, N\} \\ \int_T \sum_{k=1}^d \sum_i W_{k,i} \varphi_i \cdot \partial_k \varphi_j - \int_{\partial T} \sum_i \widehat{W}_{n,i} \varphi_i \cdot \varphi_j &= \int_T \sum_i F_i \varphi_i \cdot \varphi_j \quad \forall j \in \{1, \dots, N\}, \end{aligned} \quad (3.53)$$

$$\begin{aligned} \text{where } \int_T \sum_i F_i \varphi_i \cdot \varphi_j &= - \int_T \frac{1}{\tau} (\sum_i u_i^{m+1} \varphi_i(x) - \sum_i u_i^m \varphi_i(x))_k \cdot \varphi_j(x) dx \\ &- \int_T \sum_{l=1}^n \frac{\alpha_l}{Q_{x^l}} (X_1^l(x + \sum_i u_i^m \varphi_i(x)) - X_0^l(x)) \nabla X_1^l(x + \sum_i u_i^m \varphi_i(x))_k \cdot \varphi_j(x) dx. \end{aligned} \quad (3.54)$$

The gradient of the feature images ∇X can be calculated based on the LDG method or a different discretization method. We actually use a finite difference approximation, as it is more efficient to evaluate at points $(x + u^m(x))$ without having to determine in which element $T \in \mathcal{T}$ this point lies.

3.3.4 Implementational Details

The discretized registration method is implemented in the DUNE framework, a software library allowing the generic implementation of grid-based numerical schemes [10, 9]. The finite element implementation is based on the DUNE-FEM module [26]. Pre- and post processing, as well as input and output of images are done using ITK and VTK [39, 64]. The distance maps that represent our surfaces are calculated efficiently with the ITK implementation of the method proposed by Maurer et al. [47].

For constructing and handling the grid on which the finite element method is based, we use the ALUGRID library [27]. It supports unstructured meshes in 2D and 3D with non-conforming local adaptivity (i.e. with hanging nodes) and the possibility of domain decomposition and dynamic load balancing for parallel computations. Figure 3.8 shows a visualization of such a non-conforming locally adaptive grid. All visualizations in this thesis have been prepared with VTK and Paraview, [64, 66].

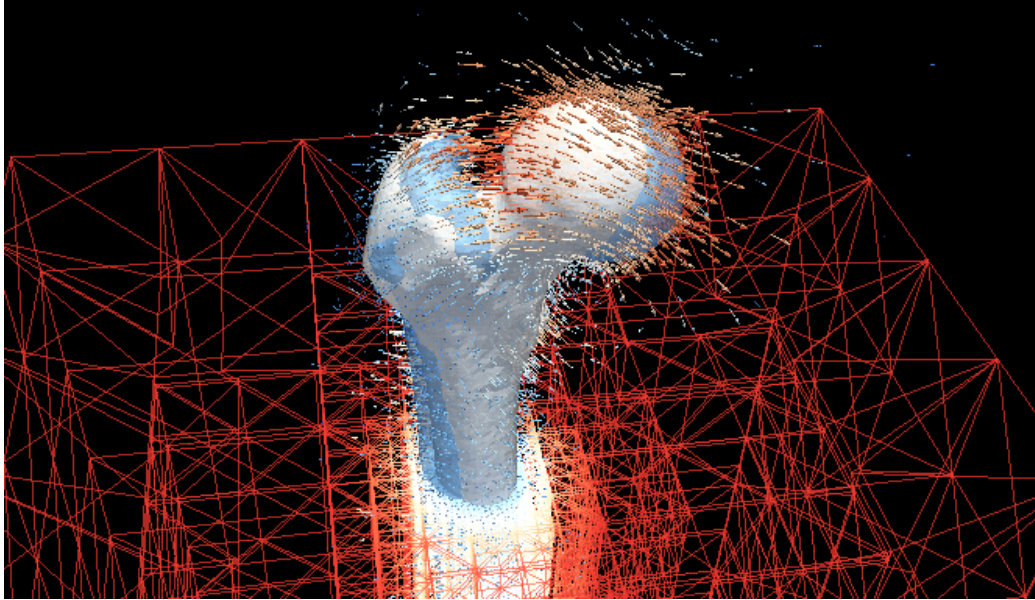


Figure 3.8: Visualization of the registration of two femurs. The deformation field that deforms one femur to be as similar to the other one as possible is calculated on an adaptive grid.

3.4 Registration Results

In this section, we show a few registration results that show how our algorithm manages to register 3D surfaces and that show the benefit of the additional terms we introduced into the registration functional. Because there is no ground truth available for the correspondence of any of our surfaces, we cannot provide any quantitative evaluation. Further, there is not a single publicly available implementation of a non-rigid surface registration algorithm that we know of, which is why we have not compared the algorithm to alternative approaches. The only thing we can provide here are qualitative examples which show the obvious improvements gained by the terms we introduced into the basic registration functional and the fact that the algorithm can in fact be used to calculate approximative correspondence between different 3D surfaces.

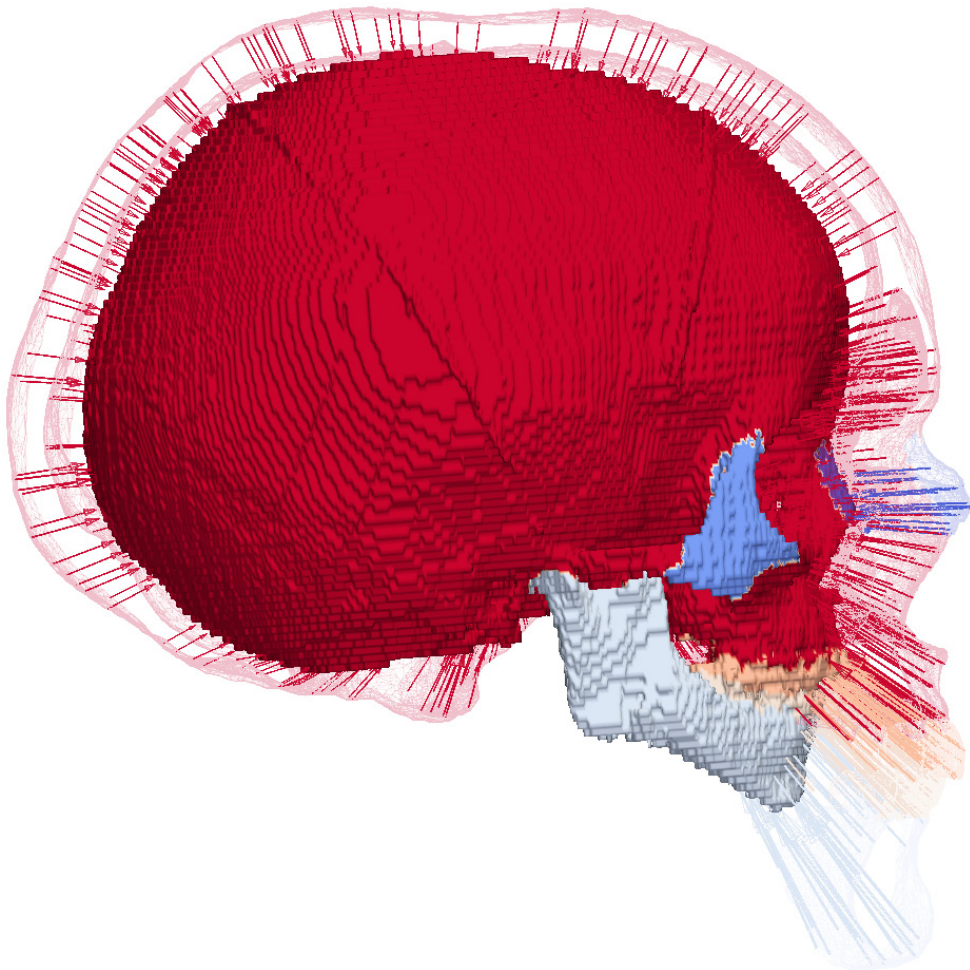


Figure 3.9: Visualization of the correspondence and the transfer of anatomical labeling between the reference (transparent outline) and a child's skull.

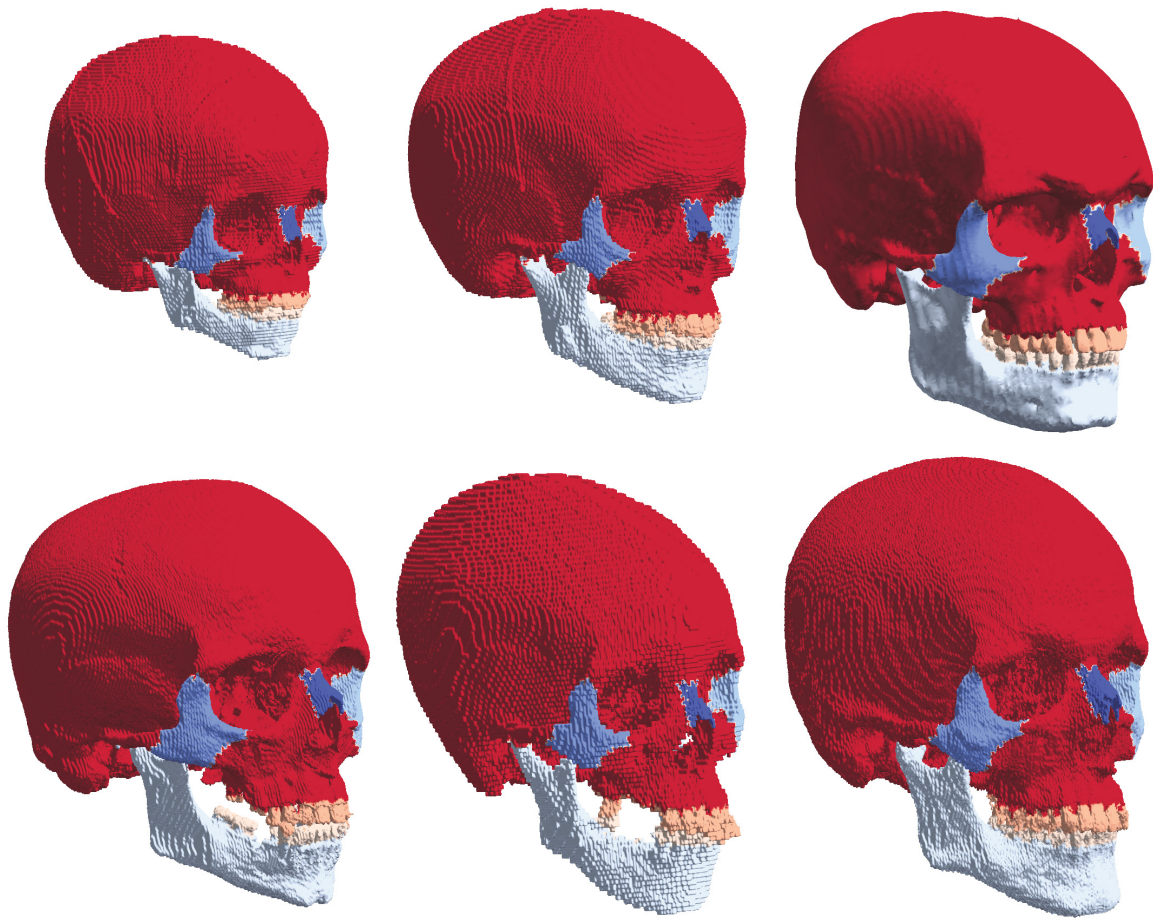


Figure 3.10: Transfer of the anatomical labelling of the reference (top right) to a variety of skulls. A transfer like this requires very accurate registration results. Even a mismatch in the number of teeth is handled properly.

3.4.1 3D Surface Registration

In this first experiment, we can observe, that thanks to the level set representation of the surfaces, our method allows the registration of two surfaces with what is essentially an image registration method. Figure 3.8 shows a visualization of this registration for two femurs on an adaptive finite element grid. Figure 3.9 shows an example of the registration of the more complicated surface of the human skull. Here, our method benefits from the fact that the level set representation is independent of the topology of the surface. The topology of the skull surface is very complicated and, due to acquisition and segmentation artifacts, not necessarily the same for two different segmented skull surfaces.

In our experiments, only the reference was anatomically labeled and hand-segmented with high attention to detail. Figure 3.9 visualizes how this labelling and segmentation can be transferred to another skull anatomy via the registration result. Figure 3.10 shows that this works over a very large range of examples from infant skulls to large adult skulls.

Even though the concept of correspondence breaks down for different numbers of teeth in the reference and the target the existing teeth are labelled correctly.

On a standard 3 GHz dual-core desktop PC with two parallel registration processes, a skull registration with 480 000 degrees of freedom (i.e. 160 000 grid points) takes about 10 minutes, a femur registration with 80 000 degrees of freedom (but more iterations) about 6 minutes. These are only indicative times to give a feeling for the run time of our algorithm. Computation times can be further reduced with more parallel processes and more aggressive parameter tuning. Note that a great advantage of our method is that the adaptive discretization requires a much inferior number of degrees of freedom than a uniform discretization. A uniform discretization with a similar resolution around the surface requires about 18 million degrees of freedom, resulting in a memory consumption of over 700 MB per deformation field.



Figure 3.11: 14 registered femurs. The original femurs are displayed as a translucent blue surface. The deformed reference is displayed with a checkerboard pattern in an attempt to visualize correspondence.

Figure 3.11 shows the registration of a larger number of femurs. The target bones are visualized as a translucent blue surface, which can hardly be seen as the registered bone, i.e. the reference deformed by the registration result, matches the target shape almost perfectly. Only some very fine details of the target bones are not matched due to the regularization

of the registration algorithm. Before applying the deformation the reference bone was colored with a checkerboard texture in an attempt to visualize correspondence. Indeed, we see that corresponding parts such as the femoral head or the trochanter major share the same texture details in all bones. Furthermore, the equal distribution of the texture over all bones shows that the registration result is smooth and free from unnatural distortions.

3.4.2 Curvature Term

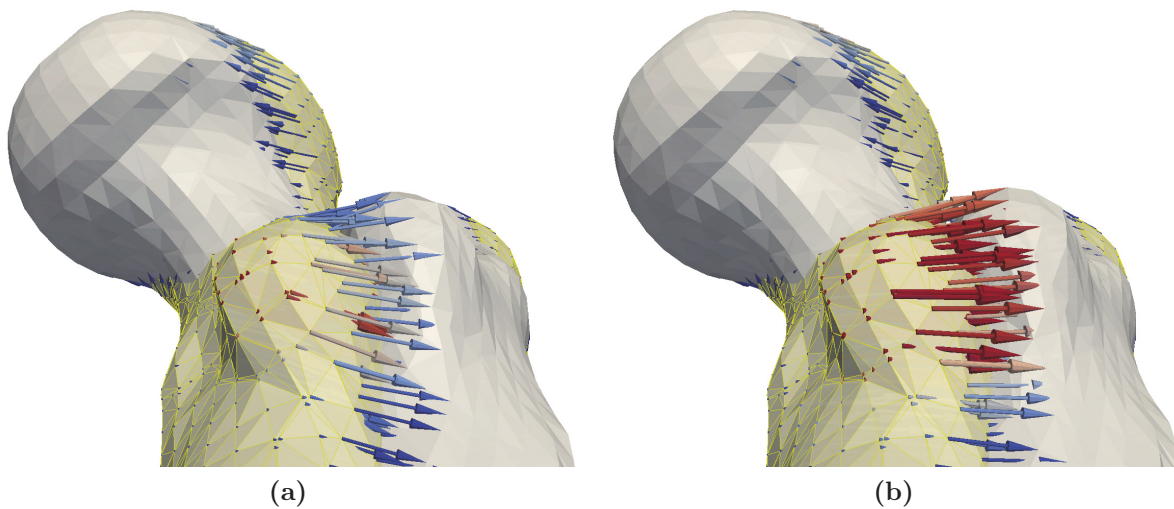


Figure 3.12: Registration of two femurs with and without curvature term. Without the curvature term, the correspondence is faulty. The corresponding features of the trochanter minor are not properly matched. The curvature term ensures a matching of corresponding shape features.

The above experiments were performed with all of the terms introduced in Section 3.1. We will now report on a couple of experiments that visualize the benefit of all additional terms, starting with the curvature term introduced in Section 3.1.5.

In Figure 3.12a, a registration between two femurs is performed without the curvature term, i.e. the surfaces are only represented by the level set functions and the original CT scans. At first glance, the surfaces are well registered, the deformation field allows us to deform the reference bone so that it is almost equal to the target bone. However, upon closer inspection, we see that the implied correspondence is faulty. The deformation field matches the top of the trochanter minor¹ of the reference to the *side* of the trochanter minor on the target. Such faulty correspondence causes problems in all subsequent applications of the registration result, such as building of statistical models or transferring anatomical labels. In Figure 3.12b, when the curvature term is used, the correspondence is much more sensible, the top of the trochanter is matched to top and the sides to the sides.

¹The trochanter minor is a prominent anatomical feature of the femur. It is the large bulge on the surfaces in Figure 3.12.

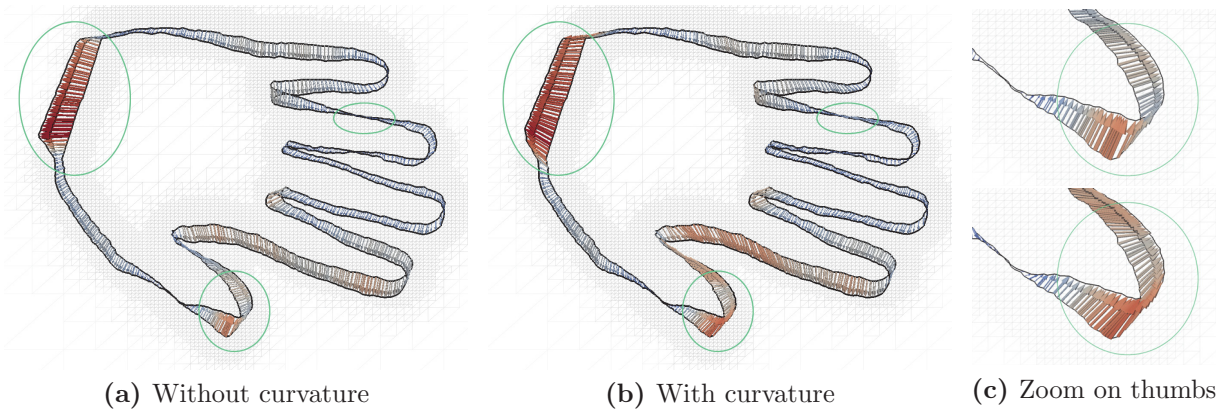


Figure 3.13: Registration of two hands with and without curvature term. With the curvature term, the defining features of the hand are matched better.

Including the mean curvature in the registration method results in superior correspondence calculation.

Figure 3.13 shows a similar example for the registration of 2D hand outlines. We have marked three areas on each hand in which the better correspondence calculation with the curvature term is apparent. On the base of the hand, where the hand shape is cut off, the registration with curvature matches corners to corners, whereas without curvature information, the corner of one hand is matched to a flat part in the other hand. On the ring finger the deformation field is almost zero at the point where the two shapes intersect, as the distance measure without curvature term is almost zero there. With the curvature term, the deformation field is not zero at this point. Instead, it naturally completes the overall deformation of the ring finger with arrows pointing to the right. Finally, on the thumb, for which we provided a magnified visualization in Figure 3.13c, the situation is similar to that of the trochanter minor in the previous example: With curvature information, the correspondence is much more sensible, properly matching the tips of the thumbs onto each other.

3.4.3 Volume Preservation

Figure 3.14a shows the reference bone warped with a deformation field of a registration without the volume preservation term. To amplify the effect for visualization purposes, we have warped the reference bone with the deformation field multiplied by 2. The coloring represents the size of the triangles that make up the surface. We see that in some places the original reference grid is unnaturally stretched resulting in very large triangles. This is the effect of large volume expansion in the deformation field. When this volume change is penalized with the volume preservation term introduced in Section 3.1.6, the resulting mesh is much more even, while still allowing an equally good matching of the target surface.

In Figure 3.14a, the weighting parameters μ and ν from Equation (3.8) have been chosen as $\mu = 2$, $\nu = 0$, i.e. no volume preservation term. In Figure 3.14b, as $\mu = \nu = 2$, i.e.

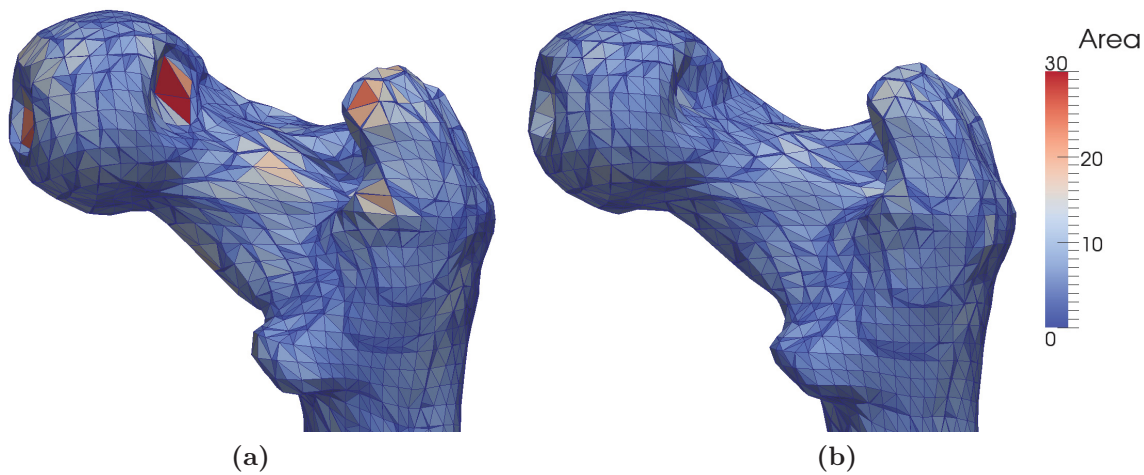


Figure 3.14: Registration of two femurs with and without volume preserving term. When the term is used, the distribution of area over the triangles of the mesh are much more even because the limited volume change prohibits strong expansion or compression of the mesh.

equal weight of the gradient and the divergence term. Simply augmenting the weight of the gradient term, e.g. $\mu = 4$, $\nu = 0$, or even $\mu = 20$, $\nu = 0$ as was done in Figure 3.15 also results in less volume change, but there is still more volume change than in Figure 3.14b, while at the same time the matching quality starts to decrease. In this example, a good trade-off between volume change and matching quality could only be reached with the use of the volume preservation term.

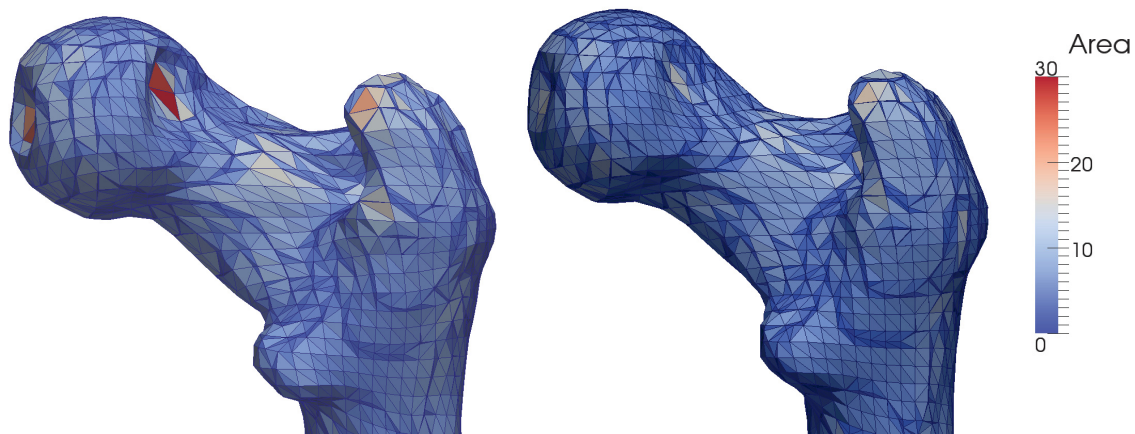


Figure 3.15: Registration with no volume preservation but increased gradient-based regularization. Large volume change is still present, while the matching quality decreases.

3.4.4 Parameters

Our proposed registration method contains many parameters that the user has to choose, many of which influence each other. Choosing the parameters optimally is an open problem, especially in the absence of ground truth data, which would provide a fair performance measure for evaluating the effect of these parameters.

Most parameters control some kind of trade-off, like that between matching accuracy and smoothness or between accuracy and computation time, and there is no definite optimum. We have managed to find a set of parameters that seemed to work well with our problem, but given a certain performance measure, it would most likely be possible to fine-tune the parameters much further.

Most parameters were chosen the same for all experiments. As the timestep size and weighting factors of all the individual terms influence each other, we kept the timestep size τ and the coefficient of the distance image α_1 fixed to 1: $\tau = \alpha_1 = 1$. We have cut off the distance image at a value of 35, i.e. 35 mm away from the surface.

When choosing the weighting factors between the feature images, we have to keep in mind that the range of the values is quite different from that of the distance image. The distance image reflects the distance from the surface in millimeters, so in our examples it typically has a range from about -20 to 40. The corresponding curvature image has a range of about -0.5 to 0.5, whereas the Hounsfield values of the CT images varies between -1000 and 3000. Therefore, the coefficient for the curvature image was chosen as $\alpha_2 = 100$ and that of the CT images as $\alpha_3 = 0.1$. Further, as the curvature and CT images are more susceptible to noise, we have used the robust Geman McClure distance measure for these, while keeping the pure L^2 measure for the distance image. These values were kept fixed for all 3D bone registrations. The regularization values were mostly kept fixed at $\mu = \nu = 4$ for most experiments, except the ones targeted specifically at exposing the properties of the regularization term in Section 3.4.3.

Regarding the discretization, we use hexahedral grids in 3D. We use a very low resolution grid with cubes of about 2 cm edge length to calculate the first coarse solution. For the next resolution level, all elements that are closer than a threshold $\theta = 35$ mm to the surface are refined, i.e. they are split in 4 hexahedra with half the edge length. In each subsequent level, the threshold for refinement is decreased by 40%, i.e. θ is replaced with 0.6θ , resulting in a grid that is heavily refined around the surface. We use 5 or 6 refinement levels, and typically run around 400 iterations of the semi-implicit iteration scheme on the lowest level. Because the initialization provided for the subsequent level in our adaptive multiresolution scheme is typically already very good, we can reduce the amount of iterations by half for each subsequent level.

Most of these parameters have been chosen based on common sense and moderate parameter-tuning. Typically, the algorithm is not sensible to small changes in the parameters. As mentioned above, automatically finding optimal parameters remains an open problem.

4 The Statistical Shape Model

In this chapter, we describe the statistical model we use to represent the shape information from the training examples. The main concept is straight-forward: the model represents shapes by linear combinations of the training examples. A probability distribution over this vector space of shapes is modeled by a normal distribution estimated from the examples.

This concept of shape modeling was described by Blanz and Vetter [13]. It relies heavily on the notion of *correspondence* and that of *linear object classes*. These are intuitive concepts that are easy to grasp but difficult or even impossible to define in a mathematically precise way.

Obviously, not all classes of shapes can be modeled by linear combinations of examples. We define a *linear object class* as a collection of shapes that can be reasonably modeled by the statistical shape models based on linear combinations described in this chapter. A necessary criterion for this is that linear or at least convex combinations of objects from the class again belong to the same class. Judging by the successful modeling by linear combinations of examples, we can assume that faces and bones are linear object classes. A combined object class including both faces and bones is clearly not a linear object class, because the linear combination between a face and a skull would not produce a useful shape. For borderline cases it is difficult to decide if a collection of shapes can be considered a linear object class or not. Tree-like structures, like vascular or bronchial trees are typical examples of such borderline cases. While for many of these structures, forming linear combinations seems possible, it is difficult or impossible to define linear combinations between structures with a different number of branches.

This brings us to the question of how the linear combination of the objects are formed. In the Morphable Model [13], shapes are represented by point sets. Each shape is represented by a finite number of points. This only makes sense if the shapes are *in correspondence*. Linear combinations should only be formed of corresponding points. For instance, in a skull model, the corner of the eye has to be calculated as a linear combination of the corners of the eyes of the examples. For shapes discretized by a finite set of points, as in [13], this means that all shapes must have the same number of points and that points with the same index or position within the set must represent corresponding points on all shapes. Figure 4.1 shows a few corresponding points for two skulls. An equivalent definition of the notion of correspondence is that every shape can be represented as a deformation of a *reference shape*. In fact, if all shapes are in correspondence, an arbitrary shape can be selected as the reference and all other shapes can be represented by deformations of this reference. This definition is actually much closer to the way our registration algorithm tries to establish correspondence: by deforming a reference to be as close as possible to a target shape. The shape model can therefore be equivalently thought of as a vector space of shapes or as a vector space of deformations.

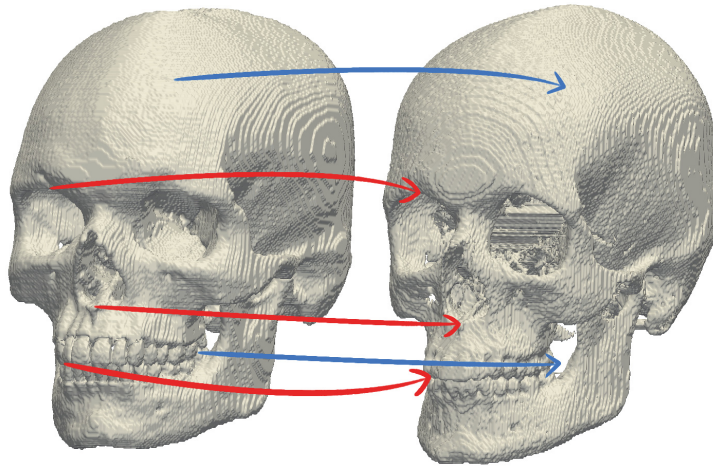


Figure 4.1: Illustrating the concept of correspondence. The red arrows mark points for which correspondence is clearly and well defined. The blue arrows mark points for which the correspondence is not so clear: On the forehead, there are many possible candidates for correspondence. For the wisdom teeth, it is not clear how correspondence can be defined if they exist in one skull but are missing in the other.

For some points like the tip of nose or the corners of the eyes, it is obvious which points correspond to each other in different shapes. For other points, which are not quite so distinctive, like a random point on the forehead it is quite hard even for a human to identify a corresponding point in other shapes of the same object class. For such featureless regions, the obvious solution, which is also what most registration algorithms do implicitly, is to define the correspondence so that it fits in smoothly with the correspondence of points representing more distinct features. When one of the shapes suffers from small defects, like the second skull from Figure 4.1, which is missing its wisdom teeth, we take the same approach: With a registration algorithm, such as the one described in the previous chapter, we try to establish correspondence as well as possible by deforming a reference shape to resemble the training example as closely as possible, and then represent the example by this deformation. In this way, even though it is almost impossible to give a precise definition of correspondence and to establish this correspondence perfectly, it is always possible to build a statistical model based on the deformations produced by a registration algorithm. Of course, the quality of the model will heavily depend on the quality of the algorithm. The model will only be as good as the registration, which underlines the outstanding importance of the registration in the construction of statistical shape models.

As the registration algorithm produces a deformation field on the entire image domain Ω around the reference shape, we can produce a number of different models all based on the same concept of correspondence, depending on what we define as the reference. Closest to the shape models introduced in [13] is the pure surface model: the reference is defined as the surface of the reference shape and the model is built from the deformation fields from the registration algorithm restricted to the reference surface. When we wish to include information about the interior of the shape, we define the complete reference

shape including its interior as the reference and build a model from its deformations. The same can be done for the outside and even the complete image domain Ω . In this way, the deformation fields produced by the registration algorithm allow us to build different models depending on the application.

In fact, the technique is not limited to modeling shapes, but can be used to model other information, such as image information captured by (medical) imaging devices, or in fact any type of data that can be reasonably represented by linear combinations of a limited number of training examples. As shape modeling is our main focus, we will usually refer to the modeled objects as “shapes”, but the reader should bear in mind that the same approach may be used for many other types of data.

In the context of shapes, every shape is represented by a deformation of the reference. Our goal in statistical shape modeling is to reflect the variability of the example shapes. We wish to find a probability distribution over the space of all deformations that models this variability. It should quantify the probability with which any given deformation belongs to the same class of objects as the examples. The probability mass of the distribution should be centered around the deformations that deform the reference into other valid shapes of the modeled object class, whereas other deformations that deform the reference into a shape that is not from the same class should have a very low probability. Obviously, there is no way to know the “true” distribution, and therefore, as in all statistical methods, we have to estimate a distribution from the example data. Without any evidence that would suggest the use of a more complicated model, we assume the most simple nontrivial distribution applicable in this scenario: the normal distribution. Of course this restricts the types of object classes we can represent with our statistical model to objects that are at least approximately normally distributed.

Typically, in the literature statistical shape models are described in discrete form. Based on the discretization method of the registration algorithm, the models are described in terms of a finite set of points or a set of spline, wavelet, or finite element coefficients. While we discretized our registration algorithm with a finite element method, it is defined continuously in terms of deformation fields $u \in L^2(\Omega, \mathbb{R}^d)$, and could be used with any other discretization method. Similarly, we wish to define the shape models not only for a specific discretization method but in a general form that can be used with any discretization method and for many possible application scenarios. Therefore, we present a formulation that introduces all of these models in a very general form, as models defined on a *Hilbert space* \mathcal{H} .

A Hilbert space \mathcal{H} is a real or complex vector space with a scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Furthermore it is a complete metric space with respect to the metric induced by the scalar product. By defining the model on a generic Hilbert space, the concept can be used for all cases mentioned above as well as for any other scenario where we wish to construct a statistical model based on training examples u_1, \dots, u_n from a Hilbert space. While the methods can surely be generalized to complex vector spaces, all our examples are real vector spaces and therefore we will assume that \mathcal{H} is a real vector space, i.e. a vector space defined of the field \mathbb{R} . We will continue by giving the most obvious choices for the Hilbert space \mathcal{H} for different application scenarios. Then, we will explain why we consider Hilbert spaces as the natural choice for defining our models.

Choices for the Hilbert Space \mathcal{H}

- For continuously defined deformation models given on an open and measurable subset $\Omega \subset \mathbb{R}^d$ such as the image domain of the reference, we choose $\mathcal{H} = L^2(\Omega, \mathbb{R}^d)$ with $\langle u, v \rangle_{\mathcal{H}} = \int_{\Omega} u(x) \cdot v(x) dx$.
- For continuously defined surface models that model different shapes as deformations of a reference surface $\Gamma \subset \mathbb{R}^d$, we have $\mathcal{H} = L^2(\Gamma, \mathbb{R}^d)$ with the scalar product defined by the surface integral $\langle u, v \rangle_{\mathcal{H}} = \int_{\Gamma} u(x) \cdot v(x) do(x)$.
- For discrete models whose training examples can be represented by vectors $\mathbf{u} \in \mathbb{R}^N$, we can choose $\mathcal{H} = \mathbb{R}^N$ with $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{H}} = \mathbf{u} \cdot \mathbf{v}$. For instance, the deformations of a reference defined by k points in \mathbb{R}^d , can be represented by vectors $\mathbf{u} \in \mathbb{R}^N = \mathbb{R}^{kd}$. If the discretization of the reference is not uniform, we can choose a different scalar product that weights each point differently.
- For any discretization method that represents the functions by a finite set of basis functions $\varphi_1, \dots, \varphi_m$, we choose $\mathcal{H} = \text{span}\{\varphi_1, \dots, \varphi_m\} \subset L^2(\Omega, \mathbb{R}^d)$ with the L^2 scalar product $\langle u, v \rangle_{\mathcal{H}} = \langle u, v \rangle_{L^2} = \int_{\Omega} u(x) \cdot v(x) dx = \sum_{i,j=1}^m u_i v_j \int_{\Omega} \varphi_i(x) \cdot \varphi_j(x) dx = \mathbf{u} \cdot \mathbf{M}\mathbf{v}$ with the matrix $(\mathbf{M})_{ij} = \langle \varphi_i, \varphi_j \rangle$. This includes functions discretized by splines, wavelets, kernels, finite elements, etc.
- For models that model not deformations but image intensities, texture or color, defined on open sets, surfaces, or point sets, the same concepts can be used by choosing d as the number of image channels. So for single-channel intensity images, such as CT images, we have $d = 1$.

Why Hilbert spaces? Hilbert spaces are the most general spaces defined in functional analysis that satisfy our requirements for building a statistical shape model similar to those introduced in [13]. In order to represent the modeled shapes by linear combinations of the training examples, the underlying space needs to be a vector space and in order to define the covariance, which describes the variation within the shape model, we need a scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. The completeness property of the Hilbert space is a prerequisite, for instance, to apply the spectral theorem, which takes the role that principal component analysis (PCA) takes in the classical statistical model [13]. Therefore, Hilbert spaces are the natural choice for defining such linear shape models in the most general setting and all models mentioned above are included in this definition.

In some ways however, the Hilbert space formulation is even too general. Some statistical results and concepts that are straight-forward in finite-dimensional spaces become extremely difficult and sometimes impossible in infinite-dimensional spaces. For instance, there is no Lebesgue measure on infinite-dimensional spaces, and therefore no canonical probability density function. Similarly, the modeling of uncorrelated Gaussian noise, which is typically used to model the deviation from the model, is much more involved in the infinite-dimensional setting.

On the other hand, most of the modeling presented in [13] and in this chapter is concerned with the compact and efficient representation of the shapes information gained from the

training examples. The actual statistical modeling as a normal distribution serves mostly as a justification for limiting the range of the model parameters to stay close to the parameters of the training examples. While some parts of this justification are not proved rigorously for the infinite-dimensional case, all resulting methods and algorithms can be transferred directly to the case of arbitrary Hilbert spaces \mathcal{H} . The actual statistical modeling is restricted to Sections 4.1.2 and 4.4.3, and some results presented there are only directly valid for finite-dimensional Hilbert spaces $\mathcal{H} \cong \mathbb{R}^n$.

One can define shape models even more generally than on Hilbert spaces. Some models defined in the literature require an even more general setting, as they define models on manifolds such as the manifold of diffeomorphisms, see [36] as a starting point. Manifolds are in general not vector spaces and therefore not Hilbert spaces. Developing statistical models on manifolds is mathematically much more involved and would rather obscure the concepts of the straight-forward statistical models employed in this thesis.

4.1 Shape Models on Hilbert Spaces

In this section we explain how to build a statistical shape model from training examples $u_1, \dots, u_n \in \mathcal{H}$ defined on a Hilbert space \mathcal{H} . While this kind of model is not limited to representing shapes but can be used for many other kinds of data, we motivate it based on the shape models we wish to construct, such as the bone models used throughout this thesis.

For shape models, the training examples u_i represent deformations of a reference shape. The model we construct represents shapes as linear combinations. In order to define a Gaussian distribution for the model in Section 4.1.2, we need to estimate the mean and covariance of the training examples. In every Hilbert space, the mean can be readily calculated as the arithmetic mean of the examples. In the finite-dimensional case, the covariance is best described by a covariance matrix, and in [13] it was shown, how a statistical shape model can be compactly and efficiently represented in terms of the eigenvectors of this covariance matrix. In the case of a model defined on a general Hilbert space \mathcal{H} , the covariance is best represented by the covariance operator, which is a linear operator $\mathcal{C} : \mathcal{H} \rightarrow \mathcal{H}$ and a direct generalization of the covariance matrix as a linear operator. In this section, we show how the mean and covariance can be estimated for training examples defined on any Hilbert space and show how the modeled shapes can be represented in terms of the eigenfunctions of the covariance operator.¹ In Section 4.1.2 we give more details on the Gaussian distribution defined by this mean and covariance.

We use the most straight-forward way to estimate mean and covariance from the training examples: the maximum likelihood estimators, also known as “sample” or “empirical” mean and covariance. The sample mean of the training examples u_i is given as:

$$\bar{u} = \frac{1}{n} \sum_{i=1}^n u_i. \quad (4.1)$$

¹We call the eigenvectors of the covariance operator “eigenfunctions” to underline the fact that \mathcal{H} is in general an infinite-dimensional function space not necessarily a finite-dimensional vector space.

The sample covariance is a symmetric bilinear form $C : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$:

$$C[v, w] = \frac{1}{n} \sum_{i=1}^n \langle v, u_i - \bar{u} \rangle_{\mathcal{H}} \langle u_i - \bar{u}, w \rangle_{\mathcal{H}} \quad (4.2)$$

With the associated *covariance operator* $\mathcal{C} : \mathcal{H} \rightarrow \mathcal{H}$:

$$\mathcal{C}[w] = \frac{1}{n} \sum_{i=1}^n (u_i - \bar{u}) \langle u_i - \bar{u}, w \rangle_{\mathcal{H}}, \quad (4.3)$$

the covariance can be expressed in terms of the Hilbert space inner product as:

$$C[v, w] = \langle v, \mathcal{C}[w] \rangle_{\mathcal{H}}. \quad (4.4)$$

If we define a linear operator $\mathcal{X} : \mathbb{R}^n \rightarrow \mathcal{H}$ by $\mathcal{X}(\mathbf{a}) = \sum_{i=1}^n a_i (u_i - \bar{u})$ and denote its Hilbert space adjoint by \mathcal{X}^* , i.e. $(\mathcal{X}^*(w))_i = \langle u_i - \bar{u}, w \rangle_{\mathcal{H}}$, we can also represent the covariance operator as:

$$\mathcal{C} =: \frac{1}{n} \mathcal{X} \mathcal{X}^*, \quad (4.5)$$

where, as is customary for linear operators and matrices, we use the notation $\mathcal{X} \mathcal{X}^* := \mathcal{X} \circ \mathcal{X}^*$ for compositions of operators. When the elements of \mathcal{H} admit a pointwise evaluation, the covariance can be defined in a pointwise fashion with the *covariance function* $c : \Omega \times \Omega \rightarrow \mathbb{R}$ defined by:

$$c(x, y) = \frac{1}{n} \sum_{i=1}^n (u_i - \bar{u})(x) (u_i - \bar{u})(y)^T. \quad (4.6)$$

The covariance operator can be expressed in terms of the covariance function as: $\mathcal{C}[w](x) = \langle c(x, \cdot), w \rangle_{\mathcal{H}}$.

In the finite-dimensional case $\mathcal{H} = \mathbb{R}^N$, the covariance is represented by the covariance matrix, and all of the different formulations of the covariance operator above represent the interpretation of the matrix as a tensor, bilinear form, operator, linear function, or 2D array of real numbers. In the general Hilbert space formulation these interpretations are given by the covariance, covariance function, and covariance operator defined above. For the most part, we will be working with the covariance operator \mathcal{C} , which will take the same central role that the covariance matrix takes in the development of the classic statistical shape models, but most concepts can be equivalently formulated in terms of the covariance function.

By definition, the covariance operator \mathcal{C} is a self adjoint Hilbert-Schmidt integral operator. Therefore it is a compact normal operator, and we can apply the spectral theorem for compact normal operators [4], which takes the place of principal component analysis performed in finite-dimensional statistical models [13]. The spectral theorem effectively provides a diagonalization or eigenvalue decomposition of the covariance operator, which greatly simplifies practically any application or theoretical consideration regarding the shape model. We present the application of the spectral theorem in the following theorem.

4.1.1 Theorem. The covariance operator \mathcal{C} defined in Equation (4.3) admits an eigenvalue decomposition with non-negative eigenvalues σ_i^2 and corresponding orthonormal eigenfunctions ρ_i . There are at most $n - 1$ nonzero eigenvalues $\sigma_1^2, \dots, \sigma_m^2$, $m < n$. \mathcal{C} can be represented as:

$$\mathcal{C}[w] = \sum_{i=1}^m \sigma_i^2 \rho_i \langle \rho_i, w \rangle_{\mathcal{H}}, \quad (4.7)$$

Moreover, the spectral theorem guarantees the orthogonal decomposition of \mathcal{H} into the span of the eigenfunctions ρ_i and the operator's kernel (= null space) $\ker \mathcal{C}$:

$$\mathcal{H} = \text{span}\{\rho_i \mid i = 1, \dots, m\} \perp \ker \mathcal{C}. \quad (4.8)$$

Proof. The proof uses results from functional analysis. See [4] for details on the theorems and concepts used.

By definition, \mathcal{C} is a Hilbert-Schmidt integral operator, and therefore compact. It is self-adjoint and therefore normal. Hence, the spectral theorem for compact normal operators applies, which proves (4.7) and (4.8). It remains to be shown that the eigenfunctions are non-negative and that $m < n$.

As $\langle w, \mathcal{C}[w] \rangle_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^n \langle u_i - \bar{u}, w \rangle_{\mathcal{H}}^2 \geq 0$, \mathcal{C} is positive semi-definite, and therefore all eigenvalues are non-negative.

It follows from Equations (4.3) and (4.7), that the range of \mathcal{C} is $\text{R}(\mathcal{C}) = \text{span}\{(u_i - \bar{u}) \mid i = 1, \dots, n\} = \text{span}\{\rho_i \mid \sigma_i^2 > 0\}$. Because $\sum_{i=1}^n (u_i - \bar{u}) = n\bar{u} - n\bar{u} = 0$, the $(u_i - \bar{u})$ are linearly dependent and therefore $\dim(\text{span}\{(u_i - \bar{u}) \mid i = 1, \dots, n\}) \leq n - 1$. Because the ρ_i are orthonormal and therefore linearly independent, it follows $\dim(\text{span}\{\rho_i \mid \sigma_i^2 > 0\}) = m \leq n - 1$. \square

The eigenvalue decomposition or diagonalization of the covariance operator provided by this theorem can also be more compactly expressed in the form of linear operators:

4.1.2 Corollary. If we define the operators $\mathcal{U}, \mathcal{Q} : \mathbb{R}^m \rightarrow \mathcal{H}$ and $\mathbf{W} \in \mathbb{R}^{m \times m}$ by:

$$\mathcal{U}(\boldsymbol{\alpha}) := \sum_{i=1}^m \alpha_i \rho_i, \quad \mathcal{Q}(\boldsymbol{\alpha}) := \sum_{i=1}^m \alpha_i \sigma_i \rho_i, \quad \mathbf{W}^2 := \text{diag}(\sigma_1^2, \dots, \sigma_m^2), \quad (4.9)$$

we have $\mathcal{C} = \mathcal{U} \mathbf{W}^2 \mathcal{U}^*$ and can represent the covariance operator \mathcal{C} as:

$$\mathcal{C} = \mathcal{U} \mathbf{W}^2 \mathcal{U}^* = \mathcal{Q} \mathcal{Q}^*, \quad (4.10)$$

and its pseudoinverse as:

$$\mathcal{C}^\dagger = \mathcal{U} \mathbf{W}^{-2} \mathcal{U}^* \quad (4.11)$$

As the eigenfunctions ρ_i are orthonormal, we have:

$$\mathcal{U}^* \mathcal{U} = \mathbf{I}_m \quad \text{and} \quad \mathcal{Q}^* \mathcal{Q} = \mathbf{W}^2. \quad (4.12)$$

\mathcal{U} is an isometry from $(\mathbb{R}^m, \langle \cdot, \cdot \rangle)$ to $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$.

Proof. Equations (4.10), (4.11) and (4.12) follow directly from Theorem 4.1.1.

We have for $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^m$:

$$\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle_{\mathbb{R}^m} = \langle \boldsymbol{\alpha}, \mathbf{I}_m \boldsymbol{\beta} \rangle_{\mathbb{R}^m} = \langle \boldsymbol{\alpha}, \mathcal{U}^* \mathcal{U} \boldsymbol{\beta} \rangle_{\mathbb{R}^m} = \langle \mathcal{U}(\boldsymbol{\alpha}), \mathcal{U}(\boldsymbol{\beta}) \rangle_{\mathcal{H}}, \quad (4.13)$$

which proves the isometry of \mathcal{U} . \square

The eigenfunctions $\{\rho_1, \dots, \rho_m\}$, which we also call ‘‘principal components’’, span an m -dimensional linear subspace of \mathcal{H} . Together with the mean \bar{u} they form the affine subspace:

$$\mathcal{M} := \bar{u} + \text{span}\{\rho_1, \dots, \rho_m\}. \quad (4.14)$$

This space, which we call the *model space* coincides with the affine space spanned by the training examples, i.e. $\mathcal{M} := \{u \in \mathcal{H} \mid u = \sum_{i=1}^n \alpha_i u_i, \sum_{i=1}^n \alpha_i = 1\} = \bar{u} + \text{span}\{u_1 - \bar{u}, \dots, u_n - \bar{u}\}$. Furthermore, we can express \mathcal{M} in terms of the operators defined above:

$$\mathcal{M} = \bar{u} + \text{img } \mathcal{X} = \bar{u} + \text{img } \mathcal{U} = \bar{u} + \text{img } \mathcal{Q}. \quad (4.15)$$

\mathcal{M} contains all affine combinations of the training examples, which are the deformations our model can represent. It is an affine subspace of \mathcal{H} . If, additionally, $0 \in \mathcal{M}$, it is also a linear subspace of \mathcal{H} . This is the case for instance in a shape model in which the reference is one of the training examples: the deformation of the reference with $0 \in \mathcal{H}$ represents the reference itself and is therefore part of the model space \mathcal{M} .

On \mathcal{M} , the covariance operator \mathcal{C} is invertible. Thanks to the eigenvalue decomposition, the inversion on \mathcal{M} is easily achieved by inverting the eigenvalues. It coincides with the formulation of the pseudo-inverse \mathcal{C}^\dagger on the whole Hilbert space \mathcal{H} :

$$\mathcal{C}^\dagger[w] = \sum_{i=1}^m \sigma_i^{-2} \rho_i \langle \rho_i, w \rangle_{\mathcal{H}} = \mathcal{U} \text{diag}(\sigma_1^{-2}, \dots, \sigma_m^{-2}) \mathcal{U}^*. \quad (4.16)$$

Each function $u \in \mathcal{M}$ in the model space can be represented by the mean \bar{u} plus a linear combination of the eigenfunctions with coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_m)$.

$$u(\boldsymbol{\alpha}) = \bar{u} + \sum_{i=1}^m \alpha_i \rho_i = \bar{u} + \mathcal{U}(\boldsymbol{\alpha}) \quad (4.17)$$

In the literature, the basis functions are often scaled by their corresponding eigenvalues σ_i^2 . Then u is represented as:

$$u(\tilde{\boldsymbol{\alpha}}) = \bar{u} + \sum_{i=1}^m \tilde{\alpha}_i \sigma_i \rho_i = \bar{u} + \sum_{i=1}^m \tilde{\alpha}_i q_i = \bar{u} + \mathcal{U}(\mathbf{W} \tilde{\boldsymbol{\alpha}}) = \bar{u} + \mathcal{Q}(\tilde{\boldsymbol{\alpha}}), \quad (4.18)$$

where $q_i := \sigma_i \rho_i$ are the scaled eigenfunctions of \mathcal{C} , and the coefficients are $\tilde{\alpha}_i = \alpha_i \sigma_i^{-1}$.

The coefficient vector $\boldsymbol{\alpha}$ resp. $\tilde{\boldsymbol{\alpha}}$ for a given deformation field $u \in \mathcal{M}$ is given by:

$$\boldsymbol{\alpha} = \mathcal{U}^*(u - \bar{u}), \quad \tilde{\boldsymbol{\alpha}} = \mathcal{Q}^*(u - \bar{u}) \quad (4.19)$$

$$\Leftrightarrow \alpha_i = \langle \rho_i, u - \bar{u} \rangle_{\mathcal{H}}, \quad \tilde{\alpha}_i = \langle \sigma_i \rho_i, u - \bar{u} \rangle_{\mathcal{H}}. \quad (4.20)$$

4.1.1 Dimensionality Reduction

PCA, which corresponds to the eigenvalue decomposition performed in Theorem (4.1.1) is often called a “dimensionality reduction” technique. Indeed, it reduces the dimensionality of the problem in two ways: Firstly, we have seen how the model, which is originally defined on the possibly infinite-dimensional Hilbert space \mathcal{H} , can be represented by m -dimensional parameter vectors $\alpha \in \mathbb{R}^m$ thanks to the basis of eigenvectors ρ_i . Of course, this could have already been achieved by using the original training examples as a basis.

The second way in which PCA can be used to reduce the dimensionality of the model is by choosing not all m eigenfunctions ρ_i of the covariance operator as a basis of the model, but only a subset. Typically, the eigenfunctions ρ_i are ordered according to the magnitude of their corresponding eigenvalues σ_i^2 and the first \tilde{m} eigenfunctions corresponding to the \tilde{m} largest eigenvalues are chosen as the basis for the model. It can easily be shown, see [11] for instance, that this is the \tilde{m} -dimensional linear model, which represents the maximum possible variance of the original data. By choosing $\tilde{m} < m$, we can reduce the dimensionality of the model while capturing as much of the variance of the data or the original model as possible.

In the following sections, we do not distinguish between \tilde{m} and m . So m can be interpreted either as being the maximum number of nonzero eigenvalues of the covariance operator or as the dimensionality of a reduced model. In the case that the dimensionality of \mathcal{H} is lower than the number of examples, which hardly ever occurs in shape modeling, m is of course be limited by the dimensionality of \mathcal{H} .

4.1.2 The Normal Distribution

In this section, we describe in more detail how the shapes are modeled statistically by a normal distribution. We have already estimated the mean \bar{u} and covariance operator \mathcal{C} from the training data. With these, we define a normal distribution $\mathcal{N}(\bar{u}, \mathcal{C})$. In the finite dimensional setting, \mathcal{C} is a covariance matrix, and $\mathcal{N}(\bar{u}, \mathcal{C})$ is a multivariate normal distribution. The generalization of the normal distribution to the infinite-dimensional setting is known as a *Gaussian process*, *Gaussian random function* or *Gaussian random field*. These are all equivalent concepts, and they are all determined completely by their mean and covariance. The covariance is typically defined by a covariance operator or a covariance function, which for our model are defined by Equation (4.3) and (4.6). Unfortunately, the study of Gaussian processes is theoretically much more involved than that of finite-dimensional multivariate normal distributions. One reason for this is that on infinite-dimensional spaces, it is not possible to define a canonical probability distribution function, because there is no Lebesgue measure with respect to which it could be defined.²

On the other hand, our model is defined only on the finite-dimensional model space

²On finite dimensional spaces, the probability density function is defined as the density (or *Radon-Nikodym derivative*) of a probability distribution with respect to the *Lebesgue measure*. On infinite dimensional spaces, it is not possible to define or construct a Lebesgue measure, and therefore not possible to define density functions with respect to a Lebesgue measure. Therefore, the concept of a canonical probability density function cannot be transferred to infinite dimensional statistics.

\mathcal{M} . On the orthogonal complement \mathcal{M}^\perp , which according to Theorem 4.1.1, coincides with the kernel of \mathcal{C} , the distribution is singular. But constrained to the model space, it is well-defined. Therefore, we consider the normal distribution $\mathcal{N}(\bar{u}, \mathcal{C})$ only on the model space. The model space \mathcal{M} is an m -dimensional affine subspace of \mathcal{H} with an orthonormal basis $\{\rho_1, \dots, \rho_m\}$. It is isomorph to \mathbb{R}^m and, regardless of the dimensionality of the embedding Hilbert space, $\mathcal{N}(\bar{u}, \mathcal{C})$ is an m -dimensional multivariate normal distribution on \mathcal{M} . Its probability density function can be expressed as:

$$p(u) = \frac{1}{\sqrt{(2\pi)^m \prod_i \sigma_i^2}} e^{-\frac{1}{2} \langle u - \bar{u}, \mathcal{C}^\dagger [u - \bar{u}] \rangle_{\mathcal{H}}}, \quad u \in \mathcal{M} \cong \mathbb{R}^m. \quad (4.21)$$

Remember that \mathcal{C}^\dagger is the pseudoinverse of \mathcal{C} , and on \mathcal{M} coincides with the inverse \mathcal{C}^{-1} . Equation (4.21) defines the probability density function in terms of elements $u \in \mathcal{M} \subset \mathcal{H}$. Via the isometry \mathcal{U} defined in Corollary 4.1.2 we can equivalently express the probability density function in terms of coefficient vectors $\boldsymbol{\alpha}$ with respect to the basis $\{\rho_1, \dots, \rho_m\}$ of \mathcal{M} . As $\mathcal{M} = \bar{u} + \text{img } \mathcal{U}$, there exists for each $u \in \mathcal{M}$ an $\boldsymbol{\alpha} \in \mathbb{R}^m$ such that: $u = \bar{u} + \mathcal{U}(\boldsymbol{\alpha})$. We can therefore express the exponent of Equation (4.21), using Corollary 4.1.2, as:

$$\begin{aligned} \langle u - \bar{u}, \mathcal{C}^\dagger [u - \bar{u}] \rangle_{\mathcal{H}} &= \langle \mathcal{U}(\boldsymbol{\alpha}), \mathcal{C}^\dagger [\mathcal{U}(\boldsymbol{\alpha})] \rangle_{\mathcal{H}} = \langle \mathcal{U}(\boldsymbol{\alpha}), \mathcal{U} \mathbf{W}^{-2} \mathcal{U}^* [\mathcal{U}(\boldsymbol{\alpha})] \rangle_{\mathcal{H}} \\ &= \langle \mathcal{U}(\boldsymbol{\alpha}), \mathcal{U} \mathbf{W}^{-2} \mathbf{I}_m \boldsymbol{\alpha} \rangle_{\mathcal{H}} = \langle \mathcal{U}(\boldsymbol{\alpha}), \mathcal{U}(\mathbf{W}^{-2} \boldsymbol{\alpha}) \rangle_{\mathcal{H}} = \langle \boldsymbol{\alpha}, \mathbf{W}^{-2} \boldsymbol{\alpha} \rangle_{\mathbb{R}^m} \end{aligned} \quad (4.22)$$

Because for the diagonal matrix $\mathbf{W}^2 = \text{diag}(\sigma_1^2, \dots, \sigma_m^2)$, we have $\det \mathbf{W}^2 = \prod_{i=1}^m \sigma_i^2$, it follows that:

$$p(u) = p(\boldsymbol{\alpha}) = \frac{1}{\sqrt{(2\pi)^m \det \mathbf{W}^2}} e^{-\frac{1}{2} \langle \boldsymbol{\alpha}, \mathbf{W}^{-2} \boldsymbol{\alpha} \rangle_{\mathbb{R}^m}}, \quad u \in \mathcal{M}, \boldsymbol{\alpha} \in \mathbb{R}^m \quad (4.23)$$

This means that in terms of the basis representation from Equation (4.17), the Gaussian distribution $\mathcal{N}(\bar{u}, \mathcal{C})$ on the model space \mathcal{M} amounts to an m -dimensional multivariate normal distribution $\mathcal{N}(0, \mathbf{W}^2)$ of the coefficient vectors $\boldsymbol{\alpha}$ with the diagonal covariance matrix \mathbf{W}^2 .

If we use the scaled eigenvectors $q_i = \sigma_i \rho_i$ as a basis for \mathcal{M} , as in Equation (4.18), we end up with an even simpler multivariate normal distribution $\mathcal{N}(0, \mathbf{I}_m)$ for the coefficient vector $\tilde{\boldsymbol{\alpha}}$ with density function:

$$p(u) = p(\tilde{\boldsymbol{\alpha}}) = \frac{1}{\sqrt{(2\pi)^m}} e^{-\frac{1}{2} \langle \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\alpha}} \rangle_{\mathbb{R}^m}}. \quad (4.24)$$

Switching between the scaled and unscaled basis is easy as a function u which is represented by the parameters $\boldsymbol{\alpha}$ in the unscaled basis is represented by the parameter vector $\tilde{\boldsymbol{\alpha}} = \mathbf{W}^{-1} \boldsymbol{\alpha}$ in the scaled basis.

While the formula for the density function given in Equation (4.21) is defined only for functions $u \in \mathcal{M}$ it can be evaluated for any function $u \in \mathcal{H}$. Note however that this does not in general constitute a well-defined density function on \mathcal{H} , because, firstly, it is singular on $\mathcal{M}^\perp = \ker \mathcal{C}$, and, secondly, on infinite dimensional spaces there is no analogon to the probability density function known from finite-dimensional statistics.

Nevertheless, the function

$$p(u) = \frac{1}{\sqrt{(2\pi)^N \prod_i \sigma_i^2}} e^{-\frac{1}{2} \langle u - \bar{u}, \mathcal{C}^\dagger [u - \bar{u}] \rangle_{\mathcal{H}}}, \quad u \in \mathcal{H} \quad (4.25)$$

can be used as a trivial extension of the density function from \mathcal{M} to \mathcal{H} .

It follows from Theorem 4.1.1 that every function $u \in \mathcal{H}$ can be decomposed into $u = u^{\mathcal{M}} + u^\perp \in \mathcal{M} \oplus \mathcal{M}^\perp$ and we have:

$$\mathcal{C}^\dagger[u] = \mathcal{C}^\dagger[u^{\mathcal{M}}] + \mathcal{C}^\dagger[u^\perp] = \mathcal{C}^\dagger[u^{\mathcal{M}}] + \sum_{i=1}^m \sigma_i^{-2} \rho_i \underbrace{\langle \rho_i, u^\perp \rangle_{\mathcal{H}}}_{=0} = \mathcal{C}^\dagger[u^{\mathcal{M}}]. \quad (4.26)$$

This means that \mathcal{C}^\dagger simply ignores u^\perp and evaluates the density of the projection of u onto the model space. In this sense, $f(u)$ is in fact only a trivial extension of the density function to \mathcal{H} . We call $f(u)$ the ‘‘pseudo likelihood function’’ of the Gaussian distribution $\mathcal{N}(\bar{u}, \mathcal{C})$ and will later use it, or rather the negative log-likelihood function

$$\frac{1}{2} \langle u - \bar{u}, \mathcal{C}^\dagger [u - \bar{u}] \rangle_{\mathcal{H}} \quad (4.27)$$

to motivate statistical regularization of several algorithms using the statistical model. In terms of the parameters $\boldsymbol{\alpha}$ resp. $\tilde{\boldsymbol{\alpha}} = \mathbf{W}^{-1} \boldsymbol{\alpha}$, the log-likelihood function reduces to:

$$\frac{1}{2} \|\mathbf{W}^{-1} \boldsymbol{\alpha}\|_{\mathbb{R}^m}^2 = \frac{1}{2} \|\tilde{\boldsymbol{\alpha}}\|_{\mathbb{R}^m}^2. \quad (4.28)$$

But as this expression ignores u^\perp , it can only be used to regularize the model component $u^{\mathcal{M}}$ of a function $u \in \mathcal{H}$ and we will have to devise a separate strategy for controlling $u^\perp \in \mathcal{M}^\perp = \ker \mathcal{C}$ in Section 4.4.

The quintessence of this section is that no matter what form the Hilbert space \mathcal{H} takes, the model is essentially described by an m -dimensional multivariate normal distribution on the model space $\mathcal{M} \subset \mathcal{H}$. The log-likelihood of a function $u \in \mathcal{M}$ can be expressed by the squared magnitude of its coefficient vector $\tilde{\boldsymbol{\alpha}}$.

4.1.3 Functional PCA

In Theorem 4.1.1, we have applied the spectral theorem for compact normal operators to prove that an eigenvalue decomposition of the covariance operator \mathcal{C} exists. Among other advantages, this decomposition allows for an efficient (pseudo-) inversion of \mathcal{C} according to Equation (4.16). For most theoretical considerations, it suffices to know that this eigenvalue decomposition exists. But obviously, for practical applications, we would like to explicitly calculate the eigenvalue decomposition of \mathcal{C} , i.e. to actually find its eigenvalues and -functions.

While the training examples and the Gaussian distribution are defined on the possibly infinite-dimensional Hilbert space \mathcal{H} , the model space \mathcal{M} is finite-dimensional. By representing the model space with a finite set of basis functions, the eigenvalue decomposition of the covariance operator can be reduced to a finite-dimensional eigenvalue decomposition. At this point, we cannot use the basis of eigenvalues ρ_i , as these are not yet known. Instead, we can naturally use the example functions $u_i - \bar{u}$ as a basis, or rather as a spanning system, as the example functions may be linearly dependent. This method for computing the eigenvalue decomposition largely follows the ideas of functional PCA [59]. The explicit expressions for the eigenfunctions and -values of our covariance operator \mathcal{C} are given in the following Lemma.

4.1.3 Lemma. Let $\sigma_1^2, \dots, \sigma_m^2$ be the $m < n$ nonzero eigenvalues of the symmetric matrix

$$\mathbf{B} := \frac{1}{n} \mathcal{X}^* \mathcal{X} \in \mathbb{R}^{n \times n} \quad \text{with components} \quad b_{ij} := \frac{1}{n} \langle u_i - \bar{u}, u_j - \bar{u} \rangle_{\mathcal{H}}. \quad (4.29)$$

Let $\mathbf{v}_i = (v_{i1}, \dots, v_{in}) \in \mathbb{R}^n$ ($i = 1, \dots, m$) be their corresponding eigenvectors. Then, the functions

$$q_i = \frac{1}{\sqrt{n}} \mathcal{X}(\mathbf{v}_i) = \frac{1}{\sqrt{n}} \sum_{j=1}^n v_{ij} (u_j - \bar{u}) \in \mathcal{H} \quad (4.30)$$

are eigenfunctions with eigenvalues σ_i^2 of the covariance operator \mathcal{C} defined in Equation (4.3). \mathcal{C} has no additional nonzero eigenvalues.

Proof. $\mathbf{B} \in \mathbb{R}^{n \times n}$ is symmetric and positive semi-definite and hence has n real non-negative eigenvalues σ_i^2 with corresponding orthonormal eigenvectors \mathbf{v}_i . As $\sum_{i=1}^n (u_i - \bar{u}) = n\bar{u} - n\bar{u} = 0$ the rows of \mathbf{B} are linearly dependent, and therefore at least one eigenvalue of \mathbf{B} is zero. Without loss of generality, the first $m < n$ eigenvalues are positive.

$$\begin{aligned} \mathcal{C}[q_i] &= \frac{1}{n} \mathcal{X} \mathcal{X}^*[q_i] = \frac{1}{n} \mathcal{X} \mathcal{X}^*\left[\frac{1}{\sqrt{n}} \mathcal{X}(\mathbf{v}_i)\right] = \frac{1}{\sqrt{n}} \mathcal{X}\left(\frac{1}{n} \mathcal{X}^* \mathcal{X} \mathbf{v}_i\right) \\ &= \frac{1}{\sqrt{n}} \mathcal{X}(\mathbf{B} \mathbf{v}_i) = \frac{1}{\sqrt{n}} \mathcal{X}(\sigma_i^2 \mathbf{v}_i) = \sigma_i^2 \frac{1}{\sqrt{n}} \mathcal{X}(\mathbf{v}_i) = \sigma_i^2 q_i \end{aligned} \quad (4.31)$$

This proves that the σ_i^2 and q_i defined above are indeed eigenvalues and -vectors of \mathcal{C} . It remains to show that there are no additional nonzero eigenvalues of \mathcal{C} . Because $\mathcal{C} = \mathcal{X} \circ \mathcal{X}^*$, we have $\text{img}(\mathcal{C}) \subset \text{img}(\mathcal{X}) = \text{span}\{u_i - \bar{u} \mid i = 1, \dots, n\}$. Since, for each eigenfunction q of \mathcal{C} , we have $\mathcal{C}[q] = \lambda q$, and therefore $\lambda q \in \text{img}(\mathcal{C})$, which implies that $\lambda = 0$ or $q \in \text{span}\{u_i - \bar{u} \mid i = 1, \dots, n\}$ and hence is of the form given in Equation (4.30), in which case the above calculation proves that λ is one of the eigenvalues σ_i^2 of \mathbf{B} . \square

This gives us a practical way to compute all nonzero eigenvalues and the corresponding eigenfunctions of the covariance operator \mathcal{C} by computing a discrete eigenvalue decomposition of the matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$. This can be achieved efficiently with a singular value decomposition of \mathbf{B} . The singular value decomposition computes a set of orthonormal eigenvectors. This means that the coefficient vectors \mathbf{v}_i are orthonormal (with respect to the Euclidean scalar product). The resulting eigenfunctions q_i are also orthogonal to each other. In order to make them orthonormal to each other with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, they need to be rescaled by their corresponding eigenvalues.

4.1.4 Lemma. Let \mathbf{B} , σ_i^2 and q_i be defined as in Lemma 4.1.3 with an orthonormal set of eigenvectors $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$. Then, the functions:

$$\rho_i = \frac{1}{\sigma_i} q_i \quad i = 1, \dots, m \quad (4.32)$$

are orthonormal eigenfunctions of the covariance operator \mathcal{C} .

Proof. As multiples of the eigenfunctions q_i , the functions ρ_i are eigenfunctions of \mathcal{C} . It remains to show their orthogonality. The orthonormal eigenvectors of \mathbf{B} diagonalize \mathbf{B} . That means:

$$\mathbf{v}_i^T \mathbf{B} \mathbf{v}_j = \sigma_i^2 \delta_{ij} \quad (4.33)$$

It follows:

$$\langle \rho_i, \rho_j \rangle_{\mathcal{H}} = \left\langle \frac{1}{\sigma_i} q_i, \frac{1}{\sigma_j} q_j \right\rangle_{\mathcal{H}} = \frac{1}{\sigma_i \sigma_j} \left\langle \frac{1}{\sqrt{n}} \mathcal{X}(\mathbf{v}_i), \frac{1}{\sqrt{n}} \mathcal{X}(\mathbf{v}_j) \right\rangle_{\mathcal{H}} \quad (4.34)$$

$$= \frac{1}{\sigma_i \sigma_j} \left\langle \mathbf{v}_i, \frac{1}{n} \mathcal{X}^* \mathcal{X}(\mathbf{v}_j) \right\rangle_{\mathbb{R}^m} = \frac{1}{\sigma_i \sigma_j} \left\langle \mathbf{v}_i, \mathbf{B} \mathbf{v}_j \right\rangle_{\mathbb{R}^m} = \frac{1}{\sigma_i \sigma_j} \sigma_i^2 \delta_{ij} = \delta_{ij}. \quad (4.35)$$

□

This means that the orthonormal eigenfunction ρ_i have the coefficients $\frac{1}{\sqrt{n\sigma_i}} \mathbf{v}_{ij}$ and:

$$\rho_i = \frac{1}{\sqrt{n\sigma_i}} \mathcal{X}(\mathbf{v}_i) = \sum_{j=1}^n \frac{1}{\sqrt{n\sigma_i}} \mathbf{v}_{ij} (u_j - \bar{u}). \quad (4.36)$$

4.1.5 Corollary. Let $\mathbf{V} \in \mathbb{R}^{n \times m}$ be the matrix whose columns are the eigenvectors \mathbf{v}_i of \mathbf{B} defined in Lemma 4.1.3. The operators $\mathcal{U}, \mathcal{Q} : \mathbb{R}^m \rightarrow \mathcal{H}$ defined in Corollary 4.1.2 can then be represented as:

$$\mathcal{Q} = \frac{1}{\sqrt{n}} \mathcal{X} \mathbf{V} \quad \text{and} \quad \mathcal{U} = \frac{1}{\sqrt{n}} \mathcal{X} \mathbf{V} \mathbf{W}^{-1}. \quad (4.37)$$

Proof. Follows directly from Lemmas 4.1.3 and 4.1.4 and Corollary 4.1.2. □

4.2 Visualization

Apart from a compact and convenient representation of the model and the reduction of its dimensionality, the eigenfunctions of the covariance operator can be used to visualize and explore the variability of the model. The eigenvalue σ_i^2 corresponding to an eigenfunction ρ_i quantifies the variation of the model, projected onto that eigenfunction. Therefore, the eigenfunction corresponding to the largest eigenvalue represents the most prominent variation of the model. The second eigenfunction represents the second most prominent variation and so on. The eigenfunctions are orthogonal to each other, and under the assumed normal distribution this means that they are stochastically independent. Each eigenfunction represents a characteristic variation of the model, which is independent of the variations of the other eigenfunctions. This analysis of the data is typically known as principal component analysis (PCA) and the eigenfunctions are referred to as principal components [11]. Usually the first few principal components represent obvious variations that can be easily put into words like the length or size of the bone. Many other components represent variations that are more difficult to describe and the last few eigenfunctions represent mostly noise.

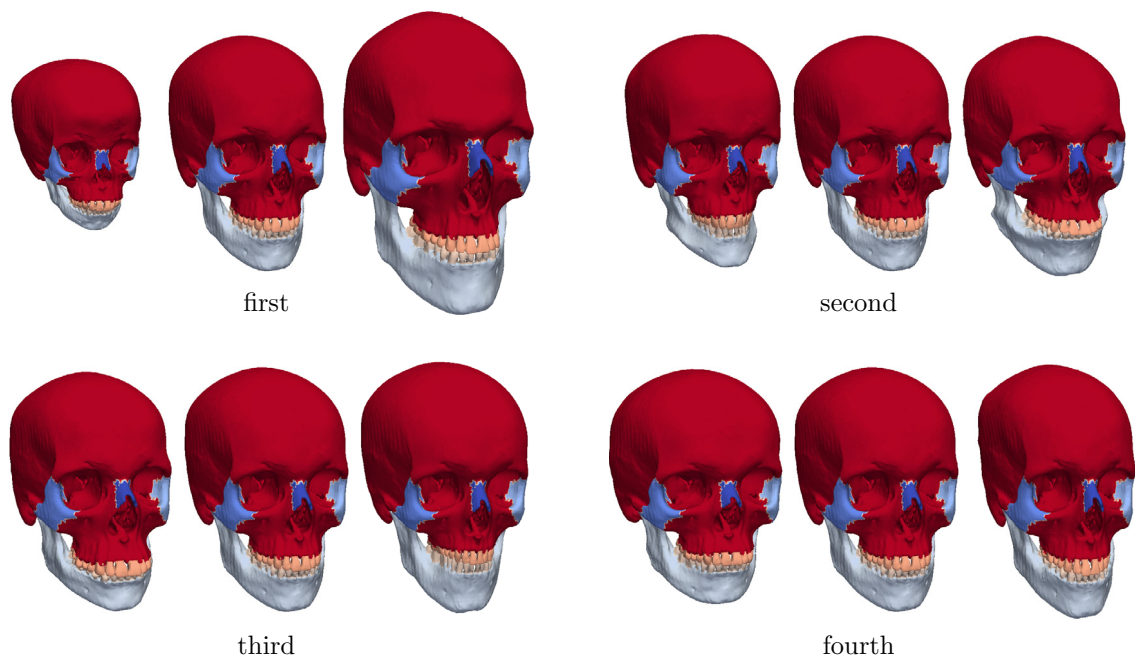


Figure 4.2: The first four principal components of a skull model. For each component, the mean is displayed in the middle, and the deformation by $\pm 3\sigma_i\rho_i$ on either side. The first component reflects the overall size of the skull. As the smallest examples skulls are skulls from infants, this overall size change is accompanied by the change of the characteristic proportions from infant to adult skulls.

In Figures 4.4, 4.3, and 4.2, we show the first few principal components of several models we constructed in the course of our work. Figure 4.3 shows our main example, the femur

model, 4.4 a model of the patella and 4.2 a model of the skull.

In all of these models, the first principal component represents mostly the overall size of the modeled bone. This confirms the first impression that the most prominent and obvious variation in bones is their size. The first principal component also shows what other variations accompany the size change in the example data. For instance, in the skull model, the smallest examples are skulls of infants and these are not only smaller overall but have a relatively large back of the head compared to the size of the face. This is reflected in the first principal component of the skull model. The following few principal components model the shape variations that are independent of size, such as width, different mandible shapes or, for the femur model, varying angles like that of the femur neck or the overall bend of the bone.

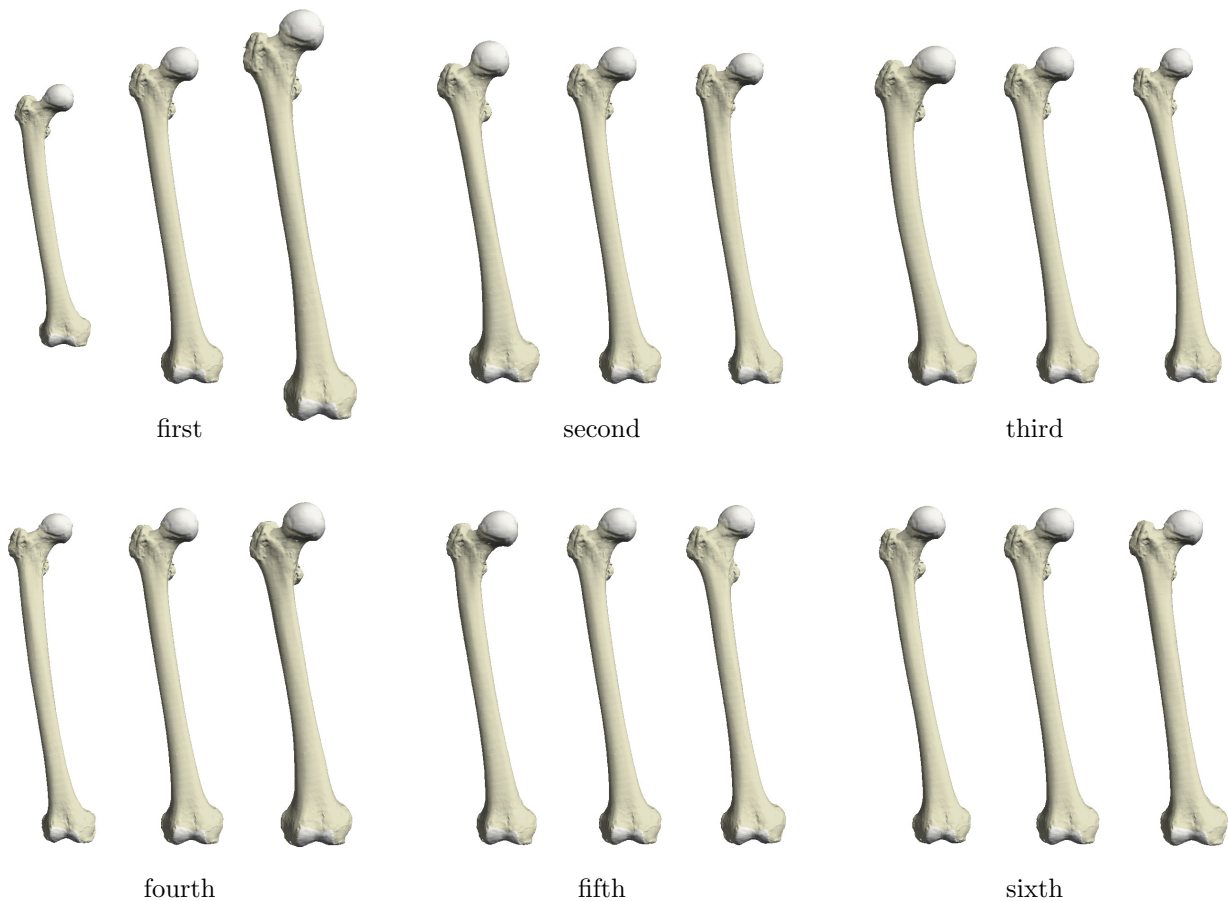


Figure 4.3: The first six principal components of our femur model. For each component, the mean is displayed in the middle, and the deformation by the eigenfunctions ρ_i on either side. The most prominent variation is the change in overall size and length exhibited by the first principal component $\pm 3\sigma_1\rho_1$. The other components reflect the variation in width, angles and bending of the femur. To make these more visible, we display the deformation by $\pm 4\sigma_i\rho_i$.

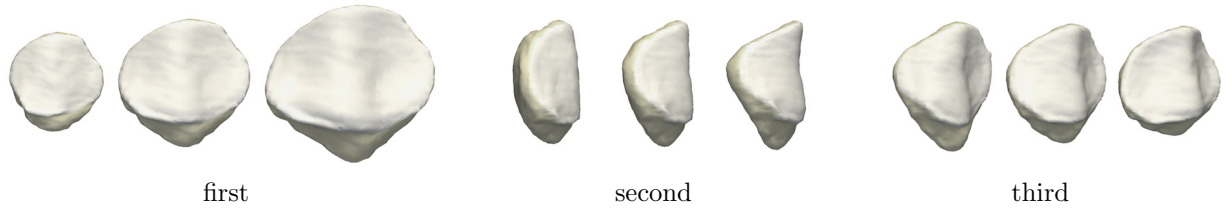


Figure 4.4: The first three principal components of a patella model. For each component, the mean is displayed in the middle, and the deformation by $\pm 3\sigma_i\rho_i$ on either side.

4.3 Discrete Models

In Section 4.1, we have proved that the covariance operator \mathcal{C} permits an eigenvalue decomposition. In Section 4.1.3, we have seen how these can be practically calculated from an eigenvalue decomposition of the matrix $\mathbf{B} = \mathcal{X}^*\mathcal{X}$. In order to actually do this, we need to assemble the matrix \mathbf{B} by computing its components

$$b_{ij} = \frac{1}{n} \langle u_i - \bar{u}, u_j - \bar{u} \rangle_{\mathcal{H}} \quad (4.38)$$

for the given examples $u_i - \bar{u}$. Depending on the Hilbert space \mathcal{H} and the training examples u_i , these components can either be calculated analytically or they have to be approximated.

Here, we will take a closer look at the case where \mathcal{H} is finite-dimensional. In any practical application, the training examples will always be given in some discrete form. Regardless of the discretization method, as a finite-dimensional real vector space, \mathcal{H} is then isomorphic to \mathbb{R}^N , where $N \in \mathbb{N}$ is the number of *degrees of freedom* (DOF) of the discretization. When formulated as a problem in \mathbb{R}^N , the construction of the statistical model can be formulated in vector and matrix form.

In the previous section, the covariance operator, on which the construction of the model is based, was defined in terms of the scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. In order to formulate the construction of the model in \mathbb{R}^N , we need to use a scalar product on \mathbb{R}^N . The most straight forward choice of scalar product in \mathbb{R}^N seems to be the standard Euclidean scalar product $\langle \cdot, \cdot \rangle_{\mathbb{R}^N}$. This scalar product is implicitly used when the model construction is defined in terms of standard matrix products as in the classical statistical models [13]. However, we will see that this is not in general the appropriate choice of scalar product for calculating the covariance of the statistical model. Instead, we propose to use the scalar product induced on \mathbb{R}^N by the Hilbert space scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. This means that we use the scalar product of the Hilbert space on which our model is defined and only express it in terms of vectors from \mathbb{R}^N .

In a finite-dimensional Hilbert space \mathcal{H} , such as the finite element space we use in our registration, each element $u \in \mathcal{H}$ can be represented as a linear combination of basis functions $\varphi_1, \dots, \varphi_N \in \mathcal{H}$:

$$u(x) = \sum_{k=1}^N u_k \varphi_k, \quad (4.39)$$

with a coefficient vector $\mathbf{u} = (u_1, \dots, u_N) \in \mathbb{R}^N$ representing the degrees of freedom (DOF) of the function u .

The scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ induces a scalar product on the space of DOF vectors \mathbb{R}^N . As all scalar products in Euclidean space, it can be represented by the Euclidean scalar product weighted with a symmetric positive definite matrix \mathbf{M} .

$$\langle u, v \rangle_{\mathcal{H}} = \langle \sum_k u_k \varphi_k, \sum_l v_l \varphi_l \rangle_{\mathcal{H}} \quad (4.40)$$

$$= \sum_{k,l=1}^N u_k v_l \langle \varphi_k, \varphi_l \rangle_{\mathcal{H}} \quad (4.41)$$

$$= \sum_{k,l=1}^N u_k v_l m_{kl} \quad (4.42)$$

$$= \mathbf{u}^T \mathbf{M} \mathbf{v} = \langle \mathbf{u}, \mathbf{M} \mathbf{v} \rangle_{\mathbb{R}^N} =: \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{M}}. \quad (4.43)$$

The matrix \mathbf{M} with elements: $m_{ij} = \langle \varphi_i, \varphi_j \rangle_{\mathcal{H}}$ is the Gram matrix of the basis vectors and is known as the “mass matrix” in the context of finite elements. It only needs to be calculated once for a given set of basis functions $\{\varphi_1, \dots, \varphi_M\}$.

Most discretization methods are based on such a basis representation and can be treated in the same way, including but not limited to splines, wavelets, Fourier approximations, and the kernel methods popular in machine learning.³ All other discretization methods can be reformulated in this way. For the standard choices of basis functions in most discretization methods, including our finite element method, the elements of \mathbf{M} can be calculated analytically.

In the discontinuous Galerkin finite element discretization we proposed for our registration method in Chapter 3, we chose orthogonal basis functions φ_i . The support of the i -th basis function is the i -th element (e.g. triangle or hexahedron) T_i of the tessellation. The elements of the mass matrix are given as: $m_{ij} = \int_{\Omega} \varphi_i \cdot \varphi_j = \delta_{ij} \int_{T_i} 1 = \delta_{ij} |T_i|$, where $|T_i|$ is the area of T_i . This means that the mass matrix \mathbf{M} is a diagonal matrix that weights the scalar product by the area of the tessellation’s elements, or in other words, by the area or volume a certain degree of freedom represents. In this way, the influence of each degree of freedom on the statistical model is weighted appropriately.

The statistical shape models [13] are described without specifying a basis $\{\varphi_1, \dots, \varphi_N\}$ and simply use the Euclidean scalar product $\langle \cdot, \cdot \rangle_{\mathbb{R}^N}$ in the construction of the model. This corresponds to choosing $\mathcal{H} = \mathbb{R}^N$ and $\{\varphi_1, \dots, \varphi_N\}$ as the standard basis with $(\varphi_i)_j = \delta_{ij}$, weighting each degree of freedom evenly.

On the other hand, these models are described on a grid or a point set and we could always construct a basis of functions $\varphi_i \in L^2(\Omega, \mathbb{R}^d)$ based on this grid, for instance using the discontinuous Galerkin basis from above. Then, the weighted scalar product $\langle \cdot, \cdot \rangle_{\mathbf{M}}$ can be used in the construction of the model. It is clear that with or without explicitly constructing such a basis it makes sense to weight each DOF by the area or volume it represents. In this light, choosing $\langle \cdot, \cdot \rangle_{\mathcal{H}} = \langle \cdot, \cdot \rangle_{\mathbb{R}^N}$ is equivalent to assuming $\mathbf{M} = \mathbf{I}_N$, which means that each DOF is weighted equally. Therefore, simply using the

³While kernel methods are typically defined as optimization problems over an infinite dimensional “Reproducing Kernel Hilbert Space” the (generalized) representer theorem proves that the solutions are always given as a finite linear combination of kernel functions [63].

Euclidean scalar product of the DOF vectors as in [13] is only justified if each degree of freedom represents roughly the same area or volume or they are uniformly distributed according to some other criterion. If the discretization is not uniform, such as in an adaptive discretization, the scalar product should be weighted accordingly, in order to ensure an appropriate influence of each degree of freedom in the model. The most straight-forward weighting scheme is weighting by the volume or area as in the discontinuous Galerkin method. In fact it is a great advantage of the discontinuous Galerkin method over other discretization methods including the standard Galerkin method that we are able to choose orthogonal basis functions resulting in a diagonal mass matrix and therefore such an easy weighting. In the following however, we do not assume that the mass matrix is diagonal. But many of the following calculations can be performed more efficiently if it is.

We have seen in Equation (4.43) that for finite-dimensional Hilbert spaces \mathcal{H} , the scalar product can be expressed as a scalar product $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{M}} = \mathbf{u}^T \mathbf{M} \mathbf{v}$ of the DOF vectors. Therefore, for discrete models, the model building can be expressed in vector and matrix notation. While this is only a reformulation of the results for general Hilbert spaces, we will quickly introduce the matrix formulation of the statistical shape models, as it may provide the key to an easy implementation. We define $\mathbf{X} \in \mathbb{R}^{N \times n}$ as the mean-subtracted data matrix, whose columns are $(\mathbf{u}_i - \bar{\mathbf{u}})$, the DOF vectors of the training examples minus the mean field $\bar{\mathbf{u}}$. With this definition, it follows from Equations (4.29) and (4.43) that:

$$\mathbf{B} = \frac{1}{n} \mathbf{X}^T \mathbf{M} \mathbf{X}. \quad (4.44)$$

The eigenvectors of \mathbf{B} can be calculated with an SVD of $\mathbf{B} = \mathbf{V}^T \mathbf{W}^2 \mathbf{V}$, with $\mathbf{W}^2 = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$ and an orthonormal matrix $\mathbf{V} = (v_{ij})$. The orthonormal eigenfunctions of the covariance operator \mathcal{C} are then, according to Lemma 4.1.4, given as:

$$\rho_i(x) = \frac{1}{\sqrt{n}\sigma_i} \sum_{j=1}^n v_{ij} (u_j - \bar{u})(x) = \frac{1}{\sqrt{n}\sigma_i} \sum_{j=1}^n v_{ij} \sum_{k=1}^N (\mathbf{u}_j - \bar{\mathbf{u}})_k \varphi_k(x) \quad (4.45)$$

$$= \sum_{k=1}^N \left(\frac{1}{\sqrt{n}\sigma_i} (\mathbf{V} \mathbf{X})_i \right)_k \varphi_k(x) \quad (4.46)$$

It follows that the DOF vector of ρ_i is given as $\boldsymbol{\rho}_i = \frac{1}{\sqrt{n}\sigma_i} (\mathbf{V} \mathbf{X})_i$. If we define the matrix $\mathbf{U} := [\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_m]$, we have:

$$\mathbf{U}^T = \frac{1}{\sqrt{n}} \mathbf{W}^{-1} \mathbf{V} \mathbf{X}^T \quad \Leftrightarrow \quad \frac{1}{\sqrt{n}} \mathbf{X} = \mathbf{U} \mathbf{W} \mathbf{V}^T. \quad (4.47)$$

The vectors $\boldsymbol{\rho}_i$ are constructed so that their associated functions are eigenfunctions of the symmetric covariance operator \mathcal{C} . The functions are orthogonal to each other in the $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ scalar product.

On the other hand, we can calculate the sample covariance matrix of the DOF vectors of the training examples as: $\boldsymbol{\Sigma} = \frac{1}{n} \mathbf{X} \mathbf{X}^T$. Note however, that the vectors $\boldsymbol{\rho}_i$ are eigenvectors of $\boldsymbol{\Sigma}$ *if and only if* the mass matrix \mathbf{M} is equal to the identity matrix \mathbf{I}_N . Also, the vectors $\boldsymbol{\rho}_i$ are orthogonal to each other in the Euclidean scalar product if and only if $\mathbf{M} = \mathbf{I}_N$.

4.3.1 Lemma. The vectors $\tilde{\boldsymbol{\rho}}_i := \mathbf{M}^{\frac{1}{2}} \boldsymbol{\rho}_i = \frac{1}{\sqrt{m\sigma_i}} \mathbf{M}^{\frac{1}{2}} (\mathbf{V}\mathbf{X})_i$ are orthogonal eigenvectors of the matrix

$$\tilde{\boldsymbol{\Sigma}} = \mathbf{M}^{\frac{1}{2}} \boldsymbol{\Sigma} \mathbf{M}^{\frac{1}{2}} = \mathbf{M}^{\frac{1}{2}} \mathbf{X} \mathbf{X}^T \mathbf{M}^{\frac{1}{2}}. \quad (4.48)$$

The corresponding eigenvectors are the σ_i^2 , i.e. the eigenvectors of the matrix \mathbf{B} and the covariance operator defined in Lemma 4.1.3.

Proof. The mass matrix $\mathbf{M} = \left(\int_{\Omega} \varphi_k \cdot \varphi_l \right)_{kl}$ is positive definite and symmetric. It therefore admits an eigenvalue decomposition $\mathbf{M} = \mathbf{S} \text{diag}(\lambda_1, \dots, \lambda_N) \mathbf{S}^T$ and a square root $\mathbf{M}^{\frac{1}{2}} = \mathbf{S} \text{diag}(\lambda_1^{\frac{1}{2}}, \dots, \lambda_N^{\frac{1}{2}}) \mathbf{S}^T$. $\mathbf{M}^{\frac{1}{2}}$ is symmetric.

With $\mathbf{U} := [\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_m]$ as above, we have:

$$\tilde{\boldsymbol{\Sigma}} = \mathbf{M}^{\frac{1}{2}} \mathbf{X} \mathbf{X}^T \mathbf{M}^{\frac{1}{2}} = \mathbf{M}^{\frac{1}{2}} \mathbf{U} \mathbf{W} \mathbf{V}^T \mathbf{V} \mathbf{W} \mathbf{U}^T \mathbf{M}^{\frac{1}{2}} = (\mathbf{M}^{\frac{1}{2}} \mathbf{U}) \mathbf{W}^2 (\mathbf{M}^{\frac{1}{2}} \mathbf{U})^T, \quad (4.49)$$

where we have used the orthonormality of \mathbf{V} . This is a diagonalization of the matrix $\tilde{\boldsymbol{\Sigma}}$ with the matrices $\mathbf{M}^{\frac{1}{2}} \mathbf{U}$ and $(\mathbf{M}^{\frac{1}{2}} \mathbf{U})^T$. In order to prove that this factorization is indeed an eigenvalue decomposition of $\tilde{\boldsymbol{\Sigma}}$, we need to prove that $(\mathbf{M}^{\frac{1}{2}} \mathbf{U})^T = (\mathbf{M}^{\frac{1}{2}} \mathbf{U})^{-1}$, i.e. that $(\mathbf{M}^{\frac{1}{2}} \mathbf{U})$ is orthonormal:

$$\begin{aligned} (\mathbf{M}^{\frac{1}{2}} \mathbf{U})^T (\mathbf{M}^{\frac{1}{2}} \mathbf{U}) &= \mathbf{U}^T \mathbf{M} \mathbf{U} = \mathbf{W}^{-1} \mathbf{V} \mathbf{X}^T \mathbf{M} \mathbf{X} \mathbf{V}^T \mathbf{W}^{-1} = \mathbf{W}^{-1} \mathbf{V} \mathbf{B} \mathbf{V}^T \mathbf{W}^{-1} \\ &= \mathbf{W}^{-1} \mathbf{V} \mathbf{V}^T \mathbf{W}^2 \mathbf{V} \mathbf{V}^T \mathbf{W}^{-1} = \mathbf{W}^{-1} \mathbf{W}^2 \mathbf{W}^{-1} = \mathbf{I}_m \end{aligned} \quad (4.50)$$

□

The vector \mathbf{u} of a deformation field represented by the model can be expressed as:

$$\mathbf{u}(\tilde{\boldsymbol{\alpha}}) = \bar{\mathbf{u}} + \sum_{i=1}^m \tilde{\alpha}_i \sigma_i \boldsymbol{\rho}_i = \sum_{i=1}^m \tilde{\alpha}_i \sigma_i \mathbf{q}_i \quad (4.51)$$

or, equivalently in matrix notation:

$$\mathbf{u}(\tilde{\boldsymbol{\alpha}}) = \mathbf{U} \mathbf{W} \tilde{\boldsymbol{\alpha}} =: \bar{\mathbf{u}} + \mathbf{Q} \tilde{\boldsymbol{\alpha}}. \quad (4.52)$$

Remember that according to Equation (4.23), under the assumption of our statistical deformation model the coefficients $\tilde{\boldsymbol{\alpha}}$ are distributed according to the m -dimensional multivariate normal distribution $\mathcal{N}(0, \mathbf{I}_m)$. Accordingly, the DOF vectors $\mathbf{u} \in \mathbb{R}^N$ are distributed according to: $\mathcal{N}(\bar{\mathbf{u}}, \mathbf{U} \mathbf{W}^2 \mathbf{U}^T) = \mathcal{N}(\bar{\mathbf{u}}, \boldsymbol{\Sigma})$. However, as pointed out before, if $\mathbf{M} \neq \mathbf{I}_N$, the matrices \mathbf{U} and \mathbf{Q} are *not* orthogonal with respect to the Euclidean scalar product, and their rows $\boldsymbol{\rho}_i$ and \mathbf{q}_i are *not* eigenvectors of $\boldsymbol{\Sigma}$.

4.3.1 Intensity Model

The model we presented in the previous sections is not limited to representing shape variations. The method can be applied to any object class that can be represented by training examples $\{u_1, \dots, u_n\}$ from a Hilbert space \mathcal{H} .

For our main example of bone models, it makes sense to build a model of the CT intensities, which represent the density of the bone tissue. CT intensities can be interpreted as functions in $L^2(\Omega)$, which is of course a Hilbert space. However, in order to form meaningful linear combinations of the intensity functions of different bones, these need to be in correspondence, so that linear combinations are taken only of the intensity of corresponding points.

As our registration results u_i from Chapter 3 are defined not only on the surface but also on the in- and outside of the objects, they can be used to bring the intensity images into correspondence. The image intensity function of the i -th example is given as a function: $g_i \in L^2(\Omega)$.

For a given point x on the reference Γ , the corresponding point in the i -th example is given as $x + u_i(x)$ and therefore the image intensity of the corresponding point as $g_i(x + u_i(x))$. The function defining the image intensity of the i -th example on the reference is given as:

$$\tilde{g}_i(x) = g_i(x + u_i(x)) \text{ for } x \in \Gamma. \quad (4.53)$$

This is a “backward warp” or “pullback” with the function $id + u_i$ of the i -th intensity function onto the reference. In many applications, we wish to consider the intensity only on a part of the image, for instance in our bone example we model only the inside of the bone. For this purpose, we can restrict the functions g_i to a subset $\check{\Gamma} \subset \Gamma$. If $\check{\Gamma}$ describes for instance the interior of the reference bone, the functions \tilde{g}_i will describe the intensity of the inside of the example bones.

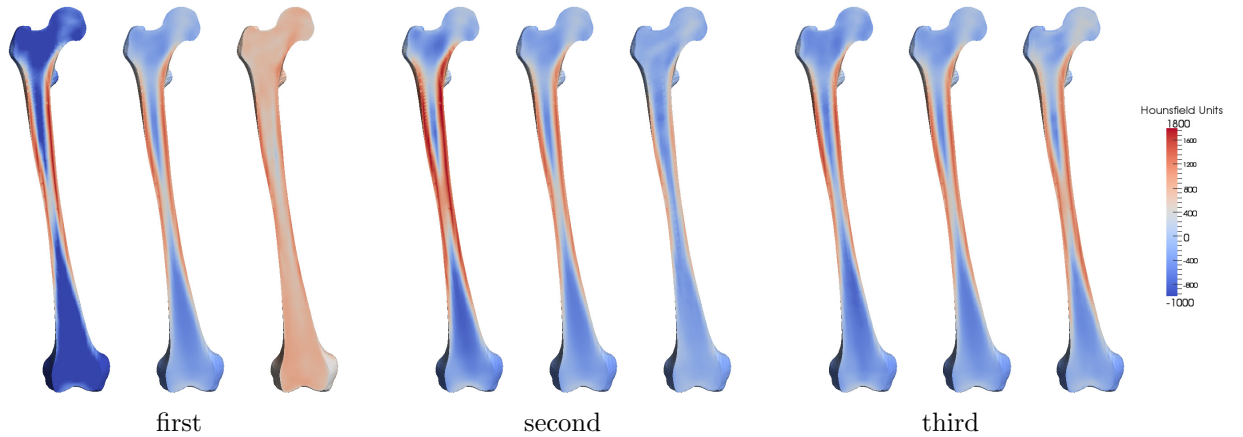


Figure 4.5: The first three principal of the femur intensity model. For each component, we display the mean shape cut in half to show the color-coded internal intensity in Hounsfield units. The mean intensity \bar{g} is displayed in the middle, and the intensity $\bar{g} \pm \sigma_{g,i} \rho_{g,i}$ to either side for each of the three components.

Consequently, we can build a statistical model like defined in Section 4.1 with the Hilbert space $\mathcal{H} = L^2(\check{\Gamma})$. In order not to confuse the shape and intensity models, we still denote the training examples of the intensity model as g_i , the modeled intensities as g , and the

coefficients β :

$$g(\beta)(x) = \bar{g}(x) + \sum_{i=1}^m \beta_i \sigma_{g,i} \rho_{g,i}(x) = \bar{g}(x) + \sum_{i=1}^m \beta_i q_{g,i}(x) \quad (4.54)$$

We used this technique to build a statistical intensity model for our femur model, which is visualized in Figure 4.5. We display the first three principal components of the intensity model and, in order to visualize the intensity inside the bone, we show a cut through the bone model. The first component represents mostly the overall density of the bone, whereas the second and third components seem to model the thickness and intensity of the bone surrounding the bone marrow canal.

While the 3D Morphable Model [13] was originally only defined *on* and not inside of the modeled 3D object and the active appearance model [22] was originally only introduced for 2D shapes, this is a straight-forward extension of these previously known models.

4.4 Dealing with \mathcal{M}^\perp

The information about the class of objects we gained from our example data sets is contained entirely in the model space \mathcal{M} , which is the affine span of the example deformations $\{u_1, \dots, u_m\}$.

On the rest of the space of deformations \mathcal{H} , more precisely on the orthogonal complement \mathcal{M}^\perp of the model space, the estimated distribution $\mathcal{N}(\bar{u}, \mathcal{C})$ is singular. That is, deformations outside the model space are assigned a probability of 0, they are modelled as impossible.

On the other hand, if we use the pseudo density function $f(u)$ defined in Equation (4.25) to evaluate the likelihood of a deformation field u , the space \mathcal{M}^\perp is ignored altogether. When we decompose a deformation field $u \in \mathcal{H}$ into $u = u^\mathcal{M} + u^\perp \in \mathcal{M} \oplus \mathcal{M}^\perp$, we have $f(u) = f(u^\mathcal{M})$. This behavior corresponds to assuming a uniform variance of ∞ on \mathcal{M}^\perp .

Obviously, neither assuming a variance of 0 or a variance of ∞ on \mathcal{M}^\perp is a very sensible way of extending the model from the model space to the space of all possible deformations. This is a well-known problem and different methods of dealing with it have been introduced in the literature. Most of them are equivalent to defining a small uniform variance of σ^2 on the space \mathcal{M}^\perp . Considering that all observed examples are elements of the model space \mathcal{M} , it makes sense to model deformations from \mathcal{M}^\perp as improbable, but not entirely impossible, which is exactly what this approach achieves.

Here, we introduce two methods that both introduce such a small variance σ^2 on \mathcal{M}^\perp . The outcome is the same for both methods, but they offer different justifications. The first, described in [62], states that it is a well known problem in statistics that for small sample sizes the sample covariance is a poor estimator of the covariance. The problem is addressed by using a “shrinkage estimator”, which is a convex combination between the sample covariance and a default covariance, such as that of white Gaussian noise.

The second method is known as PPCA and is described for instance in [11]. This interpretation and extension of the standard PCA models takes the multivariate normal

distribution $\mathcal{N}(0, \mathbf{I}_m)$ of the coefficients $\tilde{\alpha}$ from Equation (4.24) as its starting point. The model is then defined by a mapping from the space \mathbb{R}^m of coefficients to the space \mathcal{H} of deformation fields, plus white Gaussian noise with variance σ^2 . This results in a model that is identical to the model we introduced on the model space \mathcal{M} , but assumes a small variance of σ^2 on \mathcal{M}^\perp , which in this case is interpreted as Gaussian noise.

4.4.1 Shrinkage Estimation

In Equation (4.2), we have used the maximum likelihood estimator to estimate the covariance of our statistical model. However, it is well known that for small sample sizes in a high-dimensional space, this estimator performs poorly, see [62]. The main problem for our application is already described above: the estimator assigns a probability of 0 for all samples that are not in the model space. No matter how small the deviation from the model space may be, a sample that is not exactly in the span of the example used to estimate the sample covariance operator, will always be assigned a probability of 0. The result is that even new examples from the modeled object class will most likely be assigned a probability of 0.

On the other hand, if we use the pseudo-inverse of the covariance operator to evaluate the likelihood of a deformation field as proposed in Equation (4.16), we have the opposite problem: no matter how far the field is from the model space \mathcal{M} , only its projection onto \mathcal{M} is considered. In this way, shapes or deformations that are arbitrarily far away from the model space, are modeled as probable members of the object class.

Obviously, this is not the desired behavior of our model, and the problem can be attributed to the poor estimator for the covariance. The prior or default estimator of the maximum likelihood estimator is the zero covariance operator. The estimated covariance on the space on which no examples are observed, which is \mathcal{M}^\perp , is zero, resulting in a singular covariance. The standard approach to construct a nonsingular estimator is known as “shrinkage estimation” [62] and consists of taking a linear combination of the sample covariance operator \mathcal{C} and a different default covariance operator, the “shrinkage target”. The most common shrinkage target is the covariance of white Gaussian noise with variance σ^2 . White Gaussian noise has the covariance operator $\sigma^2 id$. In the finite dimensional case this corresponds to a covariance matrix $\sigma^2 \mathbf{I}_N$. The shrinkage estimator given in [62] is given as:

$$\mathcal{C}_2 = (1 - \lambda) \mathcal{C} + \lambda \sigma^2 id, \quad (4.55)$$

The parameter $\lambda \in (0, 1]$ is called the “shrinkage intensity” and determines the weight between \mathcal{C} and id . For estimating the mean, we still use the sample mean \bar{u} and therefore, our model becomes $\mathcal{N}(\bar{u}, \mathcal{C}_2)$. This distribution does not assign a variance of zero to deformation fields u outside the model space. Instead, the projection u^\perp of a vector field onto \mathcal{M}^\perp is modeled by the distribution $\mathcal{N}(0, \sigma^2 id)$. The closer u^\perp is to 0, the closer u is to \mathcal{M} , and the higher the likelihood of u^\perp and u will be. The influence of this term is controlled by the values λ and σ . These have to be chosen very small in comparison to the eigenvalues of \mathcal{C} , because functions close to the model space should naturally still be considered much more likely than functions further away. Therefore, we have $(1 - \lambda) \approx 1$

and instead of C_2 we can use a simplified estimator

$$\mathcal{C}_3 = \mathcal{C} + \sigma^2 id, \quad (4.56)$$

of course with a different σ^2 than in Equation (4.55). This estimator is also referred to as a shrinkage estimator in the literature. In [62] and other publications on shrinkage estimation, the values of σ and λ are estimated from the data, based on a more thorough statistical consideration of the problem. However, we have found that it is advantageous to leave σ^2 as a free parameter in our applications, to be chosen according to how strongly the model should weight deviations from the model space.

In our applications, we would like to explicitly evaluate the likelihood, or rather the negative log-likelihood of a deformation field $u \in \mathcal{H}$, similar to Equation (4.27). This pseudo log-likelihood is given as:

$$\mathcal{P}[u] := \frac{1}{2} \langle u - \bar{u}, \mathcal{C}_3^{-1}[u - \bar{u}] \rangle_{\mathcal{H}} \quad (4.57)$$

For this purpose, we need to calculate the inverse of covariance operator \mathcal{C}_3 . The (pseudo-) inverse of the individual operators making up \mathcal{C}_3 are \mathcal{C}^\dagger as defined in Equation (4.16) and $\sigma^{-2}id$. Both \mathcal{M} and \mathcal{M}^\perp are invariant subspaces of \mathcal{H} under \mathcal{C}_3 . Therefore, the inverse of \mathcal{C}_3 can be computed separately on \mathcal{M} and \mathcal{M}^\perp and reassembled as a full operator as:

$$\mathcal{C}_3^{-1}[w] = \sum_{i=1}^m \eta_i \rho_i \langle \rho_i, w \rangle_{\mathcal{H}} + \sigma^{-2}w. \quad (4.58)$$

with $\eta_i = (\sigma_i^2 + \sigma^2)^{-1} - \sigma^{-2}$. The calculation leading to this expression is carried out in full for the discrete case in [2]. Considering $\sigma^2 \ll \sigma_i^2$, one can choose to use the approximation $\eta_i \approx \hat{\eta}_i := \sigma_i^{-2} - \sigma^{-2}$, which corresponds to adding the shrinkage term $\sigma^2 id$ only on \mathcal{M}^\perp but not on \mathcal{M} . The log likelihood can then be represented by the compact expression:

$$\mathcal{P}[u] = \frac{1}{2} \left(\sum_{i=1}^m \hat{\eta}_i \langle \rho_i, u - \bar{u} \rangle_{\mathcal{H}}^2 + \sigma^{-2} \langle u - \bar{u}, u - \bar{u} \rangle_{\mathcal{H}} \right). \quad (4.59)$$

4.4.2 A Note on Noise

In Equations (4.55) and (4.56), we have used the covariance operator of white Gaussian noise $\sigma^2 id$ as a shrinkage target. In the finite-dimensional case, this corresponds to a multivariate Normal distribution $\mathcal{N}(0, \sigma^2 \mathbf{I}_N)$, modeling uncorrelated Gaussian noise with a uniform variance for all degrees of freedom of the model. This is the most simple and therefore most commonly used shrinkage target in the literature, see [14, 62, 25] for instance. Often its use is rationalized by assuming that it models the noise of a supposed data acquisition device, like the CT scanner. However, the finer the resolution of the model becomes, the less appropriate such a noise model becomes, because it becomes less and less likely that the noise of two adjacent points is really *independent* when the distance between points decreases. In the limit case of a model defined on an infinite-dimensional Hilbert space such as $L^2(\Omega)$, this noise model permits the independent movement even of two points that are infinitesimally close to each other. This paradoxical behavior is underlined by the fact that the covariance function corresponding to the covariance operator id is the Dirac delta function $\delta(x, y)$.

In principle, it is possible, to choose a more appropriate noise model that takes the distance between two points into account. For instance, when $\mathcal{H} = L^2(\mathbb{R})$ we could, instead of using $\mathcal{N}(0, \sigma^2 id)$ as a shrinkage target, use $\mathcal{N}(0, \sigma^2 \mathcal{K})$, where \mathcal{K} is the covariance operator defined by the covariance function:

$$k(x, y) = \frac{\tau}{2} e^{-\tau|x-y|}. \quad (4.60)$$

According to [68], this would, instead of Equation (4.59) result in the log likelihood term:

$$\mathcal{P}[u] = \frac{1}{2} \left(\sum_{i=1}^m \eta_i \langle \rho_i, u - \bar{u} \rangle_{\mathcal{H}}^2 + \sigma^{-2} \langle u - \bar{u}, u - \bar{u} \rangle_{\mathcal{H}} + \sigma^{-2} \tau^{-2} \langle \nabla(u - \bar{u}), \nabla(u - \bar{u}) \rangle_{\mathcal{H}} \right). \quad (4.61)$$

This does in fact seem like a better model as, by using the gradient, it takes the smoothness of the deformation fields $u - \bar{u}$ into account. Motivated by Equations (4.60) and (4.61), we could, slightly abusing notation, define $\mathcal{K} := id + \tau^{-2} \nabla$, and use this as a shrinkage target for our model in any Hilbert space \mathcal{H} , which would surely constitute a more realistic noise model.

Nevertheless, in this thesis we will continue to use the shrinkage target and noise model defined by the covariance operator $\sigma^2 id$, because it is the most straight-forward, results in a log likelihood function (4.59) that is very easy to evaluate, and corresponds to our own and other author's previous publications, e.g. the PPCA model presented in the next section. Furthermore, while we made a point of giving a general definition of our model for arbitrary Hilbert spaces \mathcal{H} , in the very end we will always work with discrete functions, in which case the use of the simple noise model is at least more justified. But the reader should bear in mind that this noise model is not without problems and that better alternatives exist.

4.4.3 PPCA

The probabilistic PCA model (PPCA) described in [11] offers a probabilistic interpretation for the Gaussian model including the shrinkage term introduced in the previous section. It was derived in a finite-dimensional setting. Here, we generalize the results without proof to our possibly infinite-dimensional setting. PPCA takes its starting point from the observation gained from Equation (4.24) that the model is essentially described by an m -dimensional parameter vector $\tilde{\alpha} \sim \mathcal{N}(0, \mathbf{I}_m)$. The statistical model is generative model based on the linear mapping $u : \mathbb{R}^m \rightarrow \mathcal{H}$:

$$u(\tilde{\alpha}) = \mu + \mathcal{Q}(\tilde{\alpha}) \quad \text{with} \quad \mathcal{Q}(\tilde{\alpha}) = \sum_{i=1}^m \tilde{\alpha}_i \sigma_i \rho_i. \quad (4.62)$$

In PPCA, the shrinkage target $\sigma^2 id$ used in Section 4.4.1 is interpreted as additive white Gaussian noise in this generative model. So with a noise process $\varepsilon \sim \mathcal{N}(0, \sigma^2 id)$ the PPCA model uses the mapping:

$$u(\tilde{\alpha}) = \mu + \mathcal{Q}(\tilde{\alpha}) + \varepsilon. \quad (4.63)$$

The conditional distribution of observing u given the coefficients $\tilde{\alpha}$ is:

$$p(u | \tilde{\alpha}) = \mathcal{N}(\mu + \mathcal{Q}(\tilde{\alpha}), \sigma^2 id). \quad (4.64)$$

In [11] the maximum likelihood solution for the parameters μ, w, σ^2 are computed for the finite-dimensional case. Here, we give their continuous analogons without proof:

$$\mu_{\text{ML}} = \bar{u} = \frac{1}{n} \sum_{i=1}^n u_i \quad (4.65)$$

$$\mathcal{Q}_{\text{ML}}(\tilde{\alpha})(x) = \sum_{i=1}^m \tilde{\alpha}_i (\sigma_i^2 - \sigma^2)^{\frac{1}{2}} \rho_i(x) = \sum_{i=1}^m \tilde{\alpha}_i \hat{\eta}_i^{-\frac{1}{2}} \rho_i(x) \quad (4.66)$$

$$\sigma_{\text{ML}}^2 = \frac{1}{N - m} \sum_{i=m+1}^n \sigma_i^2. \quad (4.67)$$

We see that μ_{ML} is the sample mean used in our model and \mathcal{Q}_{ML} is the linear mapping which corresponds to the likelihood function defined in (4.59).

The number of nonzero eigenvalues and eigenfunctions chosen for use in our model is denoted by $m < n$. If we choose all possible nonzero eigenvalues, the maximum likelihood parameter σ_{ML}^2 vanishes. Even if we choose to omit some eigenvalues and their eigenfunctions, σ_{ML}^2 will be very small. Therefore, as mentioned in the previous chapter, we prefer to leave σ^2 as a free parameter which has to be adapted to each individual application of the model. Furthermore, the parameter N in the formulation of σ_{ML}^2 denotes the number of degrees of freedom of the function space and is not defined in the infinite-dimensional case.

As mentioned in the previous section, a more appropriate noise model than $\varepsilon \sim \mathcal{N}(0, \sigma^2 id)$ could be chosen.

5 Applications

In this chapter, we are able to show some useful applications of the model we defined in Chapter 4. Most applications of a statistical shape or deformation model are based on at least one of these properties of the model:

- New instances of the model class can be generated by choosing coefficients $\alpha \in \mathbb{R}^m$.
- Given a shape or a deformation, we can seek the parameters α that best fit the given shape.
- The parameters α determine the *whole shape* of the modeled object. This can be used to reconstruct the complete shape from partial information.
- The log-likelihood of a shape or deformation can be computed according to Equation (4.59). This can be used to penalize unlikely shapes or deformations.

We begin by showing how this penalization of unlikely shapes and deformations with the model can be used to introduce object class specific regularization into the registration algorithm that was used to build the model in the first place.

5.1 Statistically Regularized Registration

Registration, introduced in Chapter 3, is a prerequisite to building a statistical shape model. On the other hand, the registration can itself benefit from an existing statistical model. The statistical model can be used to favor registration results that are considered likely by the model. Unlikely solutions are penalized. However, we do not want to constrain the problem strictly to the model space. Instead, we wish to allow results that are not contained in the span of the statistical model. We only want to penalize deformations that go *too far* from the model space.

Obviously, we already need a number of successful registrations in order to build the model which we then wish to use to regularize further registrations. These registrations can then profit from the “experience” gained in the previous registrations. This strategy makes most sense if we have on one hand a number of “easy” examples, such as high quality, complete, noise-free data to build the model from; and, on the other hand, “difficult” data sets whose registration can profit from the use of the statistical model. Difficult data sets may suffer from low quality/resolution, noise, incomplete data etc. The statistical model can then support the registration by allowing only results that are known to be reasonable, based on the knowledge gained from the easy registrations.

Like the other terms of the registration functional introduced in Chapter 3, the statistical model is introduced into the registration functional as an additive term. This results in a model-based regularization without a strict constraint to the model space. We introduced this idea of adding a statistical regularization term to our registration method in [2], where the term was derived for a simpler registration method whose discretization was based on an image grid.

With the general formulation of the statistical model from Chapter 4, the term representing the statistical model becomes independent of the discretization method. The statistical regularization term we add is the log-likelihood function defined in Equations (4.57) and (4.59):

$$\begin{aligned}\mathcal{P}[u] &= \frac{1}{2} \langle u - \bar{u}, \mathcal{C}_3^{-1}[u - \bar{u}] \rangle_{\mathcal{H}} \\ &= \frac{1}{2} \left(\sum_{i=1}^m \eta_i \langle \rho_i, u - \bar{u} \rangle_{\mathcal{H}}^2 + \sigma^{-2} \langle u - \bar{u}, u - \bar{u} \rangle_{\mathcal{H}} \right),\end{aligned}\quad (5.1)$$

because it is this term that quantifies the plausibility of a deformation field $u \in \mathcal{H}$. It penalizes deformations u that are unlikely according to the statistical model with distribution $\mathcal{N}(\bar{u}, \mathcal{C}_3)$. The covariance operator \mathcal{C}_3 was defined in Equation (4.56).

All necessary work for including this term was already done Chapter 4. It is also now that we profit from the general formulation of the statistical model on Hilbert spaces, as the statistical regularization term can now be included in any discrete or continuously defined registration method by simply choosing the appropriate Hilbert space \mathcal{H} . For our registration method from Chapter 3, we choose $\mathcal{H} = L^2(\Omega, \mathbb{R}^d)^1$. For use in our discretized finite element algorithm we choose $\mathcal{H} = V_h$, where V_h is the discrete finite element space endowed with the L^2 scalar product. How the term $\mathcal{P}[u]$ can be expressed for discrete deformation fields, was described in Section 4.3.

Like for the other terms of the registration functional, we need to calculate the Gateaux derivative of this term in order to include it in the minimization scheme of the registration algorithm described in Section 3.2. Taking into account that the covariance operator \mathcal{C}_3 and its inverse are linear and self-adjoint, it is easy to compute the first variation of the statistical model term \mathcal{P} as:

$$\begin{aligned}\mathcal{P}'[u, \varphi] &= \left. \frac{d}{d\varepsilon} \mathcal{P}[u + \varepsilon\varphi] \right|_{\varepsilon=0} \\ &= \left. \frac{d}{d\varepsilon} \frac{1}{2} \langle u + \varepsilon\varphi - \bar{u}, \mathcal{C}_3^{-1}[u + \varepsilon\varphi - \bar{u}] \rangle_{\mathcal{H}} \right|_{\varepsilon=0} \\ &= \frac{1}{2} \langle u - \bar{u}, \mathcal{C}_3^{-1}\varphi \rangle_{\mathcal{H}} + \frac{1}{2} \langle \varphi, \mathcal{C}_3^{-1}[u - \bar{u}] \rangle_{\mathcal{H}} \\ &= \langle \mathcal{C}_3^{-1}[u - \bar{u}], \varphi \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^m \eta_i \langle \rho_i, u - \bar{u} \rangle_{\mathcal{H}} \langle \rho_i, \varphi \rangle_{\mathcal{H}} + \sigma^{-2} \langle u - \bar{u}, \varphi \rangle_{\mathcal{H}}.\end{aligned}\quad (5.2)$$

¹Technically, we could also choose $\mathcal{H} = H_0^1(\Omega, \mathbb{R}^d)$, but this would imply the use of the more complicated H^1 scalar product.

In the implementation, this term is preceded by a weighting parameter $\gamma \in \mathbb{R}^+$ to balance its influence with the other terms of the registration method. The eigenfunctions ρ_i of the covariance operator are calculated from the discrete prior registration results as described in Section 4.3. Then, at the start of each iteration of our minimization scheme, the scalar products $\langle \rho_i, u^m - \bar{u} \rangle_{\mathcal{H}}$ can be calculated for the current solution u^m . The rest of the statistical term \mathcal{P} can be treated implicitly in our semi-implicit iteration scheme from Section 3.3.1.

Our choice of orthogonal finite element basis functions φ_i mentioned in Section 3.3.2, which results in a diagonal mass matrix \mathbf{M} makes the calculation of the eigenfunctions ρ_i relatively easy and straight-forward. This choice of basis functions was made possible by the use of the local discontinuous Galerkin discretization method.

Prior Work

The concept of statistical deformation models and their inclusion into registration algorithms has been previously researched by several groups [32, 73, 61, 75]. However, except for the method of Wang and Staib [73], these methods either constrain the registration result strictly to the span of the model or perform the statistical regularization in a separate step. Our statistical regularizer is integrated into our registration method and, unlike all of the previous methods, is formulated independently of the discretization method.

5.1.1 Registration Results

In these experiments we demonstrate the use and benefit of using prior knowledge in the form of a statistical model in our registration method. To build the statistical model, we first performed registrations of 15 intact hands for the 2D example and 45 intact femurs for the 3D example. From these, statistical deformation models are built as described in Chapter 4. These models are then used as prior knowledge for registering difficult or damaged data sets. In Figure 5.1, this prior knowledge allows the registration of the reference to a hand with a missing ring finger. Here, we would like to compute a registration result that registers this hand as if it were complete. That is, the shape of the hand should be well registered where it is intact, while the missing finger should be filled in by the prior knowledge from the statistical model. Without this prior knowledge, in Figure 5.1a, the ring finger of the reference is deformed unnaturally in an attempt to match the hand with the missing finger. Note that the use of a robust distance measure and the regularization terms prevent a complete disappearance or distortion of the finger. When the prior knowledge from the statistical deformation model is used in Figure 5.1b, the hand is well-matched but the model enforces a registration with an intact ring finger, as all its training data sets include 5 intact fingers.

Note that this result shows a full registration and not merely a fit of the model. This means that the registration result is not strictly constrained to the model space \mathcal{M} of the statistical model, but only “encouraged” by the penalty term from Equation 5.1 not to lie too far from the model space. For our example this means that the statistical regularization term does not permit a registration result with a missing or strongly distorted ring finger.

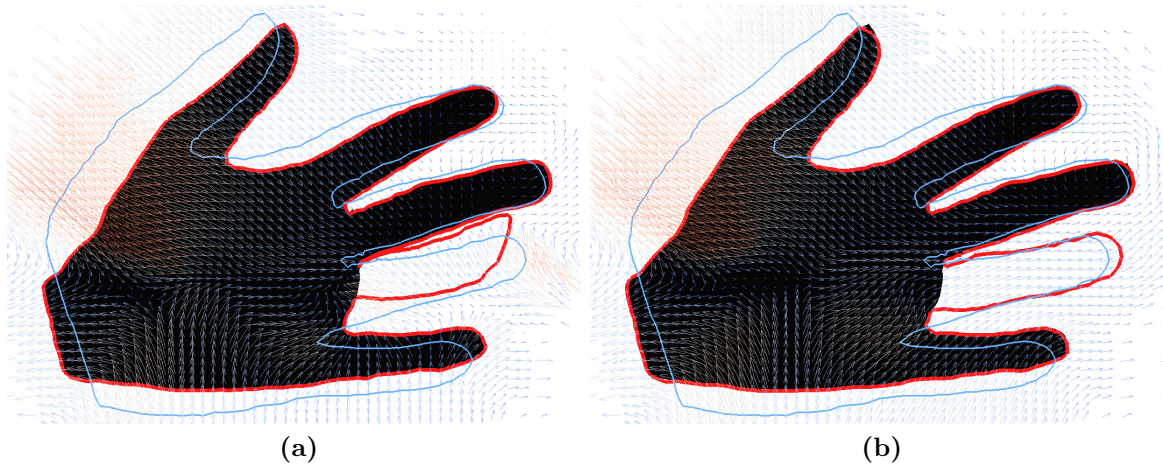


Figure 5.1: The reference shape (blue line) is registered onto a hand with a missing finger. The red line shows the warp of the reference with the resulting deformation field, without (a) and with statistical regularization (b).

At the same time, it still accurately registers the intact rest of the hand even if it does not strictly lie in the model space. In this way, we can capture the individual characteristics of this specific hand. When we add this result as an additional example into our statistical model, we will actually add new information. If we had constrained the problem strictly to the model space, we would have only gained information that is already included in the model.

In Figure 5.2 a similar experiment is shown for 3D bone registration. The prior knowledge from 45 previous registrations is used in the registration of two damaged femurs. The first femur has an artificial hip joint prosthesis and the second one a missing trochanter major. The registration method with statistical regularization registers the complete bones while complementing the missing parts from its prior knowledge. In these experiments, we have used 40 modes of variation (principal components) of the model. In a uniform discretization as that used in [2] with a similar resolution around the surface this registration would have required 28 GB of memory.

These experiments were performed with a robust distance measure, cf. Section 3.1.3. In the places with missing data, the regular L^2 measure has a very large value and would dominate the registration process including the static regularization term.

As with all regularization techniques, a balance between over- and under-regularization has to be found. If too little weight is placed on the static regularization term, it fails to prevent unnatural deformations and we end up with registrations that are too far from the model space. With too much weight, the result is pulled too close to the model, and we are essentially only performing model fitting.

Model fitting methods, which fit the statistical model to data by searching for an instance of the model strictly within the model space are introduced in the next section.

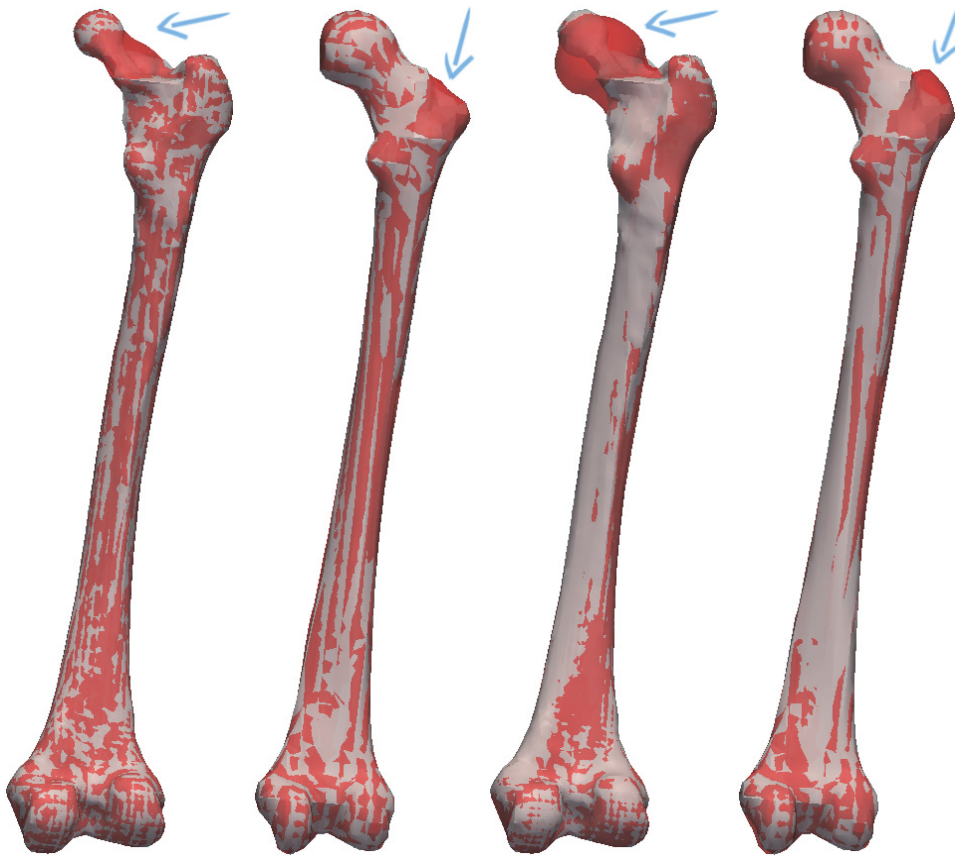


Figure 5.2: Registration of a pair of damaged bones, with and without statistical regularization. The first bone has an artificial hip joint and in the second, the trochanter major is missing. Without statistical regularization the damaged bones are matched exactly (left two images). With statistical regularization, the method recognizes that the damaged parts do not conform with the prior knowledge and fills in the missing information from the model (right two images).

5.2 Statistical Shape Model Fitting

In the previous section, we have introduced a way to regularize a registration method with a statistical deformation model. The registration result was purposely not strictly confined to the model space, in order to allow us to capture shape variations that are not yet part of the given model. This is an important property if we wish to include the registration result as an additional example into the model.

However, in other scenarios this is not necessary and we can search for shapes that best fit a given data set strictly within the model space. The advantage of this approach is that instead of having to calculate a complete deformation field with a (finite element) registration algorithm, we only need to find a parameter vector $\alpha \in \mathbb{R}^m$ of the model, in order to fit the model to a data set. Additionally, with model fitting, it is possible to fit the model to more types of data than with registration. The statistical model can be fitted to many different types of data including but not limited to landmark points, (partial)

surfaces, and complete medical images.

In model fitting we search for shapes that fit this given data only among the model instances $u \in \mathcal{M}$, which can be represented as:

$$u(\boldsymbol{\alpha}) = \bar{u} + \sum_{i=1}^m \alpha_i \rho_i. \quad (5.3)$$

In this Chapter, we assume that all functions in \mathcal{H} can be evaluated pointwise. For the sake of readability, we define $u_{\boldsymbol{\alpha}} := u(\boldsymbol{\alpha})$ and denote the evaluation of a model instance at a point $x \in \mathbb{R}^d$ by:

$$u_{\boldsymbol{\alpha}}(x) := u(\boldsymbol{\alpha})(x) = \bar{u}(x) + \sum_{i=1}^m \alpha_i \rho_i(x). \quad (5.4)$$

Different algorithms have to be used for different combinations of model and data. In order to not repeat the same ideas several times, we first introduce the concepts that apply to all different ways of model fitting:

Model Instances In a shape model, u represents the deformation of a reference $\Gamma \subset \mathbb{R}^d$, $d = 2, 3$. We denote the reference deformed by the deformation field $u_{\boldsymbol{\alpha}}$ by:

$$\Gamma_{\boldsymbol{\alpha}} = \{x + u_{\boldsymbol{\alpha}}(x) \mid x \in \Gamma\}. \quad (5.5)$$

In principle, the model can be defined on any set $\Gamma \subset \mathbb{R}^d$, but in practice Γ is usually either a surface, an open subset of \mathbb{R}^d , or a discrete point set representing a shape. Similarly, the data we wish to fit to can be given in many different forms. Often, it is a subset of \mathbb{R}^d , but one can even imagine more general types of data such as functions or probability distributions.

Distance Measures All methods we present here are formulated as minimization problems. A distance measure is defined that measures the difference between the model and the data. By minimizing this measure with respect to the model parameters $\boldsymbol{\alpha}$, we hope to find an instance of the model that matches the given data as well as possible.

We will denote the data by the symbol \mathfrak{X} , the distance measure will be denoted as a functional $\mathcal{D}[\mathfrak{X}, \Gamma_{\boldsymbol{\alpha}}]$. It quantifies the distance between the model and the data set as a nonnegative real number and typically has to be designed for each application and type of data set. Fitting the statistical model to the data is then defined as the minimization problem of finding the model parameters $\boldsymbol{\alpha} \in \mathbb{R}^m$ that minimize the distance measure:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^m}{\operatorname{argmin}} \mathcal{D}[\mathfrak{X}, \Gamma_{\boldsymbol{\alpha}}]. \quad (5.6)$$

In the context of this minimization problem, the data set \mathfrak{X} and the model Γ are fixed. Only the parameters $\boldsymbol{\alpha}$ can be changed to minimize the distance measure. Therefore, we can interpret \mathcal{D} as a function:

$$\mathcal{D}(\boldsymbol{\alpha}) : \mathbb{R}^m \rightarrow \mathbb{R}_0^+. \quad (5.7)$$

In order to minimize \mathcal{D} with respect to $\boldsymbol{\alpha}$, we usually have to calculate the gradient $\nabla\mathcal{D}(\boldsymbol{\alpha})$, which, due to the chain rule, involves taking the derivative of the deformation $u_{\boldsymbol{\alpha}}$ with respect to $\boldsymbol{\alpha}$. The partial derivatives of the deformations, which appear in all fitting methods, take a very simple form:

$$\partial_{\alpha_j} u_{\boldsymbol{\alpha}}(x) = \sum_{i=1}^m \partial_{\alpha_j} \alpha_i \rho_i(x) = \sum_{i=1}^m \delta_{ij} \rho_i(x) = \rho_j(x). \quad (5.8)$$

Spatial Transformation As the model is built from example data sets that are aligned to the reference, the model can only represent shape changes, but not changes to the position of the model in \mathbb{R}^d . Therefore, if we cannot assume that the data set we wish to fit our model to is perfectly aligned with the reference, we can introduce an additional spatial transformation $\mathbf{T}_{\boldsymbol{\theta}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ with parameters $\boldsymbol{\theta}$ to transform the model's position in space. Depending on the application, $\mathbf{T}_{\boldsymbol{\theta}}$ can for instance be a rigid transform of the form $\mathbf{T}_{\boldsymbol{\theta}}x = \mathbf{A}x + \mathbf{b}$ or a similarity transformation of the form $\mathbf{T}_{\boldsymbol{\theta}}x = \lambda\mathbf{A}x + \mathbf{b}$, in which case the parameter vector $\boldsymbol{\theta}$ represents the rotation matrix \mathbf{A} , the translation vector \mathbf{b} , and the scaling factor λ . The deformed and transformed surface is then given as:

$$\Gamma_{\boldsymbol{\alpha}, \boldsymbol{\theta}} = \{\mathbf{T}_{\boldsymbol{\theta}}(x + u_{\boldsymbol{\alpha}}(x)) \mid x \in \Gamma\}. \quad (5.9)$$

The distance measure \mathcal{D} becomes a function $\mathcal{D}(\boldsymbol{\alpha}, \boldsymbol{\theta})$ of the model and transformation parameters $(\boldsymbol{\alpha}, \boldsymbol{\theta})$. The additional spatial transformation can be regarded as integrating a spatial alignment into the fitting process. Depending on the specific application and model, this alignment can be performed prior to, after, or concurrently with the fitting. For instance, if landmark information is available the data and model can be aligned before fitting as described in Section 2.3. In any case, as the spatial transformation does not provide any valuable insight into the statistical model fitting and can be considered a relatively easy problem that only makes the following methods and formulae more tedious and less readable, we assume, with *virtually no loss of generality*, that the data set \mathfrak{X} and the model $\Gamma_{\boldsymbol{\alpha}}$ are already aligned and we only need to optimize with respect to the parameters $\boldsymbol{\alpha}$ and work with distance measures of the form $\mathcal{D}(\boldsymbol{\alpha})$.

Regularization We try to minimize distance measure $\mathcal{D}(\boldsymbol{\alpha})$ with an iterative minimization algorithm or, when possible, analytically. The result is a parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ for which the distance measure attains at least a local minimum. It is however not guaranteed that $\boldsymbol{\alpha}$ represents a valid shape of the object class modeled by the shape model.

From Section 4.1.2 we know that under the assumed normal distribution of our model, the model parameters $\boldsymbol{\alpha}$ are distributed according to a multivariate normal distribution $\mathcal{N}(0, \mathbf{W}^2)$ with negative log-likelihood function:

$$f(\boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{W}^{-1}\boldsymbol{\alpha}\|_{\mathbb{R}^m}^2 = \frac{1}{2} \|\tilde{\boldsymbol{\alpha}}\|_{\mathbb{R}^m}^2. \quad (5.10)$$

This means that coefficient vectors with high $f(\boldsymbol{\alpha})$ have a low value of the probability density function $p(\boldsymbol{\alpha})$ defined in Equation (4.23) and are considered unlikely by our model.

Therefore, by adding the penalty term $\mathcal{R}[\boldsymbol{\alpha}] = \lambda f(\boldsymbol{\alpha})$ to the distance measure, we discourage unlikely results, in order to allow only valid members of the object class as possible results. The weighting term $\lambda \in \mathbb{R}^+$ determines the balance between the distance measure and this regularization term $\mathcal{R}[\boldsymbol{\alpha}]$. The gradient of the regularization term is:

$$\nabla \mathcal{R}[u_{\boldsymbol{\alpha}}] = \mathbf{W}^{-2} \boldsymbol{\alpha}. \quad (5.11)$$

If the distance measure $\mathcal{D}(\boldsymbol{\alpha})$ is minimized with an iterative optimization algorithm with a small stepsize and an initial guess of $\boldsymbol{\alpha} = 0$, it is often not necessary to use the regularization term, because the algorithm typically never reaches local minima $\boldsymbol{\alpha}$ far away from the initial guess 0 that would have a large value $\mathcal{R}[\boldsymbol{\alpha}]$.

Least Squares While many other types of distance measures are imaginable, all measures we represent here are defined as integrals or, for discrete point sets, as sums over the points of the model $\Gamma_{\boldsymbol{\alpha}}$ or the data \mathfrak{X} . Even more specifically, we introduce all of our fitting methods with least squares distance measures. This means that all distance measures are of the form:

$$\mathcal{D}(\boldsymbol{\alpha}) = \frac{1}{2} \int f(\boldsymbol{\alpha})^2 \quad \text{or} \quad \mathcal{D}(\boldsymbol{\alpha}) = \frac{1}{2} \sum_i f(\boldsymbol{\alpha})_i^2 \quad (5.12)$$

with some function f . It is well known that such least squares type distance measures are only appropriate for relatively noise-free data. When the data sets we wish to fit our model to exhibit excessive noise, the least squares measure can be replaced with a more robust distance measure. In this sense, the least squares measures introduced here should be seen only as a suggestion for one of many possible distance measures.

5.2.1 Fitting with known Correspondence

We begin with the easiest case of model fitting: the case in which the correspondence between the data \mathfrak{X} and the model $\Gamma_{\boldsymbol{\alpha}}$ is known a priori. With given correspondence information, we already know that the data can be represented as a deformation of the reference Γ , or a subset $\check{\Gamma} \subset \Gamma$. This means that we already have a deformation field $u : \check{\Gamma} \rightarrow \mathbb{R}^3$ such that

$$\mathfrak{X} = \{x + u(x) \mid x \in \check{\Gamma}\}. \quad (5.13)$$

In this case, the fitting problem is reduced to finding $\boldsymbol{\alpha}$ that $u_{\boldsymbol{\alpha}}$ is as close as possible to u . In general u is not contained in the model space \mathcal{M} and therefore we have to find an approximative solution $u_{\boldsymbol{\alpha}} \approx u$. We calculate the least squares approximation by minimizing the distance measure

$$\mathcal{D}(\boldsymbol{\alpha}) = \frac{1}{2} \|u - u_{\boldsymbol{\alpha}}\|^2. \quad (5.14)$$

\mathcal{D} is a convex function, with a unique minimum at $\nabla \mathcal{D} = 0$. The components of the gradient $\nabla \mathcal{D}(\boldsymbol{\alpha})$ are given as:

$$(\nabla \mathcal{D}(\boldsymbol{\alpha}))_j = \partial_{\alpha_j} \mathcal{D}(\boldsymbol{\alpha}) = \langle u - u_{\boldsymbol{\alpha}}, \partial_{\alpha_j} u_{\boldsymbol{\alpha}} \rangle = \langle u - u_{\boldsymbol{\alpha}}, \rho_j \rangle. \quad (5.15)$$

It follows

$$\nabla \mathcal{D}(\boldsymbol{\alpha}) = 0 \quad \Leftrightarrow \quad \langle u, \rho_j \rangle = \langle u_{\boldsymbol{\alpha}}, \rho_j \rangle \quad \forall j \in \{1, \dots, m\} \quad (5.16)$$

$$\Leftrightarrow \quad \langle u, \rho_j \rangle = \sum_{i=1}^m \alpha_i \langle \rho_i, \rho_j \rangle = \sum_{i=1}^m \alpha_i \delta_{ij} \quad \forall j \quad (5.17)$$

$$\Leftrightarrow \quad \alpha_j = \langle u, \rho_j \rangle \quad \forall j, \quad (5.18)$$

i.e. the least squares approximation of u is simply the orthogonal projection of u onto the model space $\mathcal{M} = \text{span}\{\rho_1, \dots, \rho_m\}$.

If we add the regularization term $\mathcal{R}[\boldsymbol{\alpha}]$, we have:

$$\nabla \mathcal{D}(\boldsymbol{\alpha}) + \mathcal{R}(\boldsymbol{\alpha}) = 0 \quad \Leftrightarrow \quad \alpha_j = \langle u, \rho_j \rangle - \lambda \sigma_j^{-2} \alpha_j \quad \forall j \in \{1, \dots, m\}. \quad (5.19)$$

Here, we can see clearly, how the regularization term \mathcal{R} penalizes large model coefficients α_j .

This is a very straight-forward and simple method to find the parameters $\boldsymbol{\alpha}$ such that the deformation field $u_{\boldsymbol{\alpha}}$ best fits the given field u in a least squares sense. If we consider the probabilistic PCA model introduced in Section 4.4.3, we can calculate not only a single shape $u_{\boldsymbol{\alpha}}$ but a full probability distribution of the parameters $\boldsymbol{\alpha}$ given the deformation field u . This is of particular interest if u is defined only on a part of the reference $\check{\Gamma}$ as it allows us to investigate how this part influences the full shape.

Note that both for this and the next section, the fitting of the model to a partial shape specifically includes the case of the partial shape $\check{\Gamma}$ consisting of a set of *isolated landmark points*.

5.2.2 Reconstruction of Partial Shapes

In the previous section we have calculated the model coefficients that best fit a given data set. The result is a parameter vector that describes the single best-fitting model, as judged by the distance measure \mathcal{D} and regularization term \mathcal{R}

In many applications, there may be many possible model instances that fit the given data equally well, especially if we wish to fit the model to a partial data set \mathfrak{X} . For instance, when \mathfrak{X} is only a part of the bone. In this case we can fit the full model to the partial data, thus using the model to reconstruct the partial bone to its full shape. However, we may wish to identify or model all of the possible reconstructions and not just the single one that fits the data best and/or is judged as the most probable by our model. Considering all possible reconstruction also gives us an indication of how reliable the proposed best reconstruction is.

Ideally, we would like to model this with the conditional distribution of the model parameters $\boldsymbol{\alpha}$ given the data \mathfrak{X} . This can in principle be computed using the normal distribution $\mathcal{N}(\bar{u}, \mathcal{C})$ of our model. However, because our model is built from a finite number of training examples, usually in the range between 20 and 200, the dimension of the parameter vector $\boldsymbol{\alpha}$ is usually quite limited. On the other hand, the data \mathfrak{X} is often of much higher, theoretically even of infinite dimensionality. Therefore even when only a

small part of the shape is given, the problem of finding the distribution of α given the data \mathfrak{X} is usually over-constrained. In other words, the distribution of α given \mathfrak{X} is singular. This is partly due to the fact that even the slightest details and even noise present in the data \mathcal{X} can completely determine and even over-constrain the fitting of the model to the data.

By using the Probabilistic PCA model defined in Section 4.4.3, which introduced a small variance in addition to the model, we can permit a certain amount of noise and small inaccuracies in the fitting. This allows us to formulate a nonsingular conditional distribution for the parameters α given the data \mathfrak{X} , which models all variations of the model that fit the given data reasonably well. These are, for instance, all bone shapes that, up to a small error, fit a given partial bone. The conditional distribution will be close to singular only if the data \mathfrak{X} completely determines the shape up to the additional flexibility introduced by the PPCA model. This is the case, for instance, if \mathcal{X} represents the whole shape.

The PPCA model is formulated in terms of the parameters $\tilde{\alpha} = \mathbf{W}^{-1}\alpha$, which are distributed according to $\mathcal{N}(0, \mathbf{I})$. The conditional distribution of a deformation u given the parameters $\tilde{\alpha}$ is, according to Section 4.4.3:

$$p(u | \tilde{\alpha}) = \mathcal{N}(\bar{u} + \mathcal{Q}(\tilde{\alpha}), \sigma^2 id), \quad (5.20)$$

with $\mathcal{Q}(\tilde{\alpha}) = \sum_{i=1}^m \tilde{\alpha}_i \sigma_i \rho_i$. As in the previous section, we assume that the given data \mathfrak{X} is in correspondence with a subset $\check{\Gamma} \subseteq \Gamma$. The PPCA model for the partial model defined on $\check{\Gamma}$ takes the same form but with:

$$\mathcal{Q}(\tilde{\alpha}) = \sum_{i=1}^m \tilde{\alpha}_i \sigma_i \check{\rho}_i, \quad (5.21)$$

using the basis functions restricted to $\check{\Gamma}$: $\check{\rho}_i := \rho_i|_{\check{\Gamma}}$.

As the data set \mathfrak{X} is in correspondence with the model, it can be represented by a deformation of $\check{\Gamma}$ by a deformation field u . The conditional distribution of the model parameters $\tilde{\alpha}$ given the deformation u can be computed by Bayes Rule:

$$p(\tilde{\alpha} | u) = \frac{p(u | \tilde{\alpha})p(\tilde{\alpha})}{p(u)}. \quad (5.22)$$

In the PPCA model, the deformation field is an affine transformation of the model parameters $\tilde{\alpha}$. In this simple case of a *linear Gaussian model*, the conditional distribution $p(\tilde{\alpha} | u)$ is a Gaussian distribution whose mean and covariance can be computed explicitly and take the form given in the following Proposition:

5.2.1 Proposition (Bayes' Rule for Linear Gaussian Models). Let \mathcal{G} and \mathcal{H} be two Hilbert spaces. Let x be a random function on \mathcal{G} and y a random function on \mathcal{H} . Given a marginal Gaussian distribution for x and a conditional distribution for $y | x$ in the form:

$$p(x) = \mathcal{N}(\mu, \Lambda) \quad (5.23)$$

$$p(y | x) = \mathcal{N}(A(x) + b, L), \quad (5.24)$$

with a linear operator $A : \mathcal{G} \rightarrow \mathcal{H}$ and covariance operators $\Lambda : \mathcal{G} \rightarrow \mathcal{G}$ and $L : \mathcal{H} \rightarrow \mathcal{H}$. Let $A^* : \mathcal{H} \rightarrow \mathcal{G}$ be the Hilbert space adjoint to A . Then, the marginal distribution of y and the conditional distribution of x given y are given by:

$$p(y) = \mathcal{N}(A(\mu) + b, L + A\Lambda A^*) \quad (5.25)$$

$$p(x|y) = \mathcal{N}(\Sigma A^* L^{-1}(y - b) + \Lambda^{-1}\mu, \Sigma) \quad (5.26)$$

where

$$\Sigma = (\Lambda + A^* L A)^{-1}. \quad (5.27)$$

Proof. This result is proved for $\mathcal{G} = \mathbb{R}^m, \mathcal{H} = \mathbb{R}^n$ in Section 2.3.3 of [11]. \square

This result was proved in [11] only for the case $\mathcal{G} = \mathbb{R}^m, \mathcal{H} = \mathbb{R}^n$. The proof relies heavily on the use of the probability density function (w.r.t. the Lebesgue measure) of the multivariate normal distribution, which we know does not exist in the infinite-dimensional case. A proof in a more general case would have to be carried out in terms of Gaussian processes or Gaussian measures. We do not attempt to give a proof in a more general case, but are convinced that the result holds at least for our PPCA model, where we have $\mathcal{G} = \mathbb{R}^m$ and \mathcal{H} the Hilbert space in which our model is defined. For the PCA model we have:

$$p(\tilde{\alpha}) = \mathcal{N}(0, \mathbf{I}) \quad (5.28)$$

$$p(u | \tilde{\alpha}) = \mathcal{N}(\mathcal{Q}(\tilde{\alpha}) + \bar{u}, \sigma^{-2} id), \quad (5.29)$$

and hence, by Proposition 5.2.1:

$$p(\tilde{\alpha}|u) = \mathcal{N}(\mu_{\tilde{\alpha}|u}, \Sigma_{\tilde{\alpha}|u}), \quad (5.30)$$

$$\text{with } \Sigma_{\tilde{\alpha}|u} = (\mathbf{I} + \sigma^{-2} \mathcal{Q}^* \mathcal{Q})^{-1}, \quad (5.31)$$

$$\mu_{\tilde{\alpha}|u} = \Sigma_{\tilde{\alpha}|u} \mathcal{Q}^* (\sigma^{-2}(u - \bar{u})). \quad (5.32)$$

As for the PPCA model, $\mathcal{G} = \mathbb{R}^m$, the covariance operator $\Sigma_{\tilde{\alpha}|u}$ is an $m \times m$ -dimensional covariance matrix. The operator $\mathcal{Q}^* \mathcal{Q}$ is matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$ with components $s_{ij} = \langle \sigma_i \check{\rho}_i, \sigma_j \check{\rho}_j \rangle_{\mathcal{H}}$. The covariance matrix $\Sigma_{\tilde{\alpha}|u}$ is:

$$\Sigma_{\tilde{\alpha}|u} = (\mathbf{I} + \sigma^{-2} \mathbf{S})^{-1}. \quad (5.33)$$

If $\check{\Gamma} = \Gamma$, we have $\check{\rho}_i = \rho_i$ and because the ρ_i are orthonormal, $\mathbf{S} = \mathbf{W} = \text{diag}(\sigma_i^2)$. Otherwise, \mathbf{S} has to be calculated. Using the notation from Section 4.3 and defining $\check{\mathbf{U}}$ as the matrix of the DOF vectors of $\check{\rho}_i$, \mathbf{S} can be calculated for discrete functions as:

$$\mathbf{S} = \check{\mathbf{U}}^T \mathbf{M} \check{\mathbf{U}}. \quad (5.34)$$

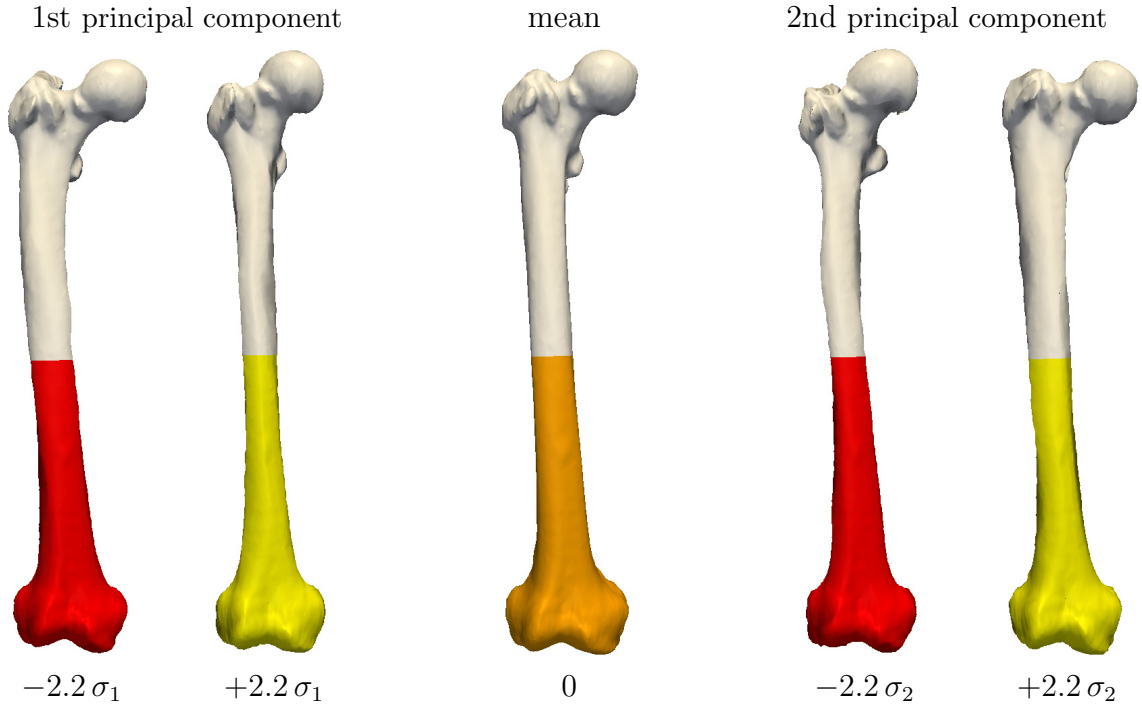


Figure 5.3: The two first principal components model of the flexibility model of a femur in which the distal (colored, bottom) part is fixed.

5.2.3 Flexibility Model

The conditional distribution $p(\tilde{\alpha}|u)$ defined above describes the parameters $\tilde{\alpha}$ of all shapes that, up to small deviations or noise, fit the deformation u which represents the given data set \mathfrak{X} . For a parameter vector $\tilde{\alpha}$, the shape is given by deforming the reference by the deformation:

$$u(\tilde{\alpha}) = \bar{u} + \mathcal{Q}(\tilde{\alpha}). \quad (5.35)$$

Depending on whether we use the full basis functions ρ_i or the restricted ones $\check{\rho}_i$ the full or the partial shape is reconstructed. Note that in the PPCA model noise modeled by $\varepsilon \sim \mathcal{N}(0, \sigma^2 id)$ is added to this shape, see Equation (4.63). However, for reconstructing the most probable shape for a given parameter vector $\tilde{\alpha}$ we do not add additional noise, which equates to adding 0, the mean of the noise model.

The conditional distribution $p(\tilde{\alpha}|u)$ implies a different distribution on the deformation fields in \mathcal{H} than the original one based on the distribution $p(\tilde{\alpha}) = \mathcal{N}(0, \mathbf{I})$. The distribution of the deformation fields whose parameters are distributed according to $p(\tilde{\alpha}|u)$ is the normal distribution $\mathcal{N}(\mu_{\tilde{\alpha}|u}, \Sigma_{\tilde{\alpha}|u})$ transformed by \mathcal{Q} :

$$\mathcal{N}(\mathcal{Q}(\mu_{\tilde{\alpha}|u}), \mathcal{Q}\Sigma_{\tilde{\alpha}|u}\mathcal{Q}^*). \quad (5.36)$$

In the previous section, we have presented the expressions for \mathcal{Q} , $\mu_{\tilde{\alpha}|u}$, and $\Sigma_{\tilde{\alpha}|u}$ in terms of the eigenfunctions ρ_i of the original model's covariance operator. For several purposes, for instance the visualization of the typical deformations of the model described

by $\mathcal{N}(\mathcal{Q}(\mu_{\hat{\alpha}|u}), \mathcal{Q}\Sigma_{\hat{\alpha}|u}\mathcal{Q}^*)$, it is advantageous to calculate a new set of basis functions consisting of the eigenvalues of the covariance operator $\mathcal{Q}\Sigma_{\hat{\alpha}|u}\mathcal{Q}^*$. This can be achieved with the methods of functional PCA proposed in Section 4.1.3.

Remember that in Section 4.1.3, the covariance operator could be expressed as $\mathcal{C} = \frac{1}{n}\mathcal{X}\mathcal{X}^*$. Therefore, by replacing \mathcal{X} with $\sqrt{n}\mathcal{Q}\Sigma_{\hat{\alpha}|u}^{1/2}$, the complete procedure of building a new model can be carried out as described in Section 4.1.3 for this new covariance operator. This amounts to using the training examples $\sqrt{n}\sum_{k=1}^m\Sigma_{\hat{\alpha}|u_{ki}}^{1/2}\rho_k$, $i = 1, \dots, m$ instead of $u_i - \bar{u}$ in the construction of the model. The matrix \mathbf{B} , whose eigenvalue decomposition was used to practically compute the eigenvalue decomposition of the covariance operator is for the new model $\mathbf{B} = \Sigma_{\hat{\alpha}|u}^{1/2}\mathcal{Q}^*\mathcal{Q}\Sigma_{\hat{\alpha}|u}^{1/2}$, and can be equally easily calculated. This method was first published in [44].

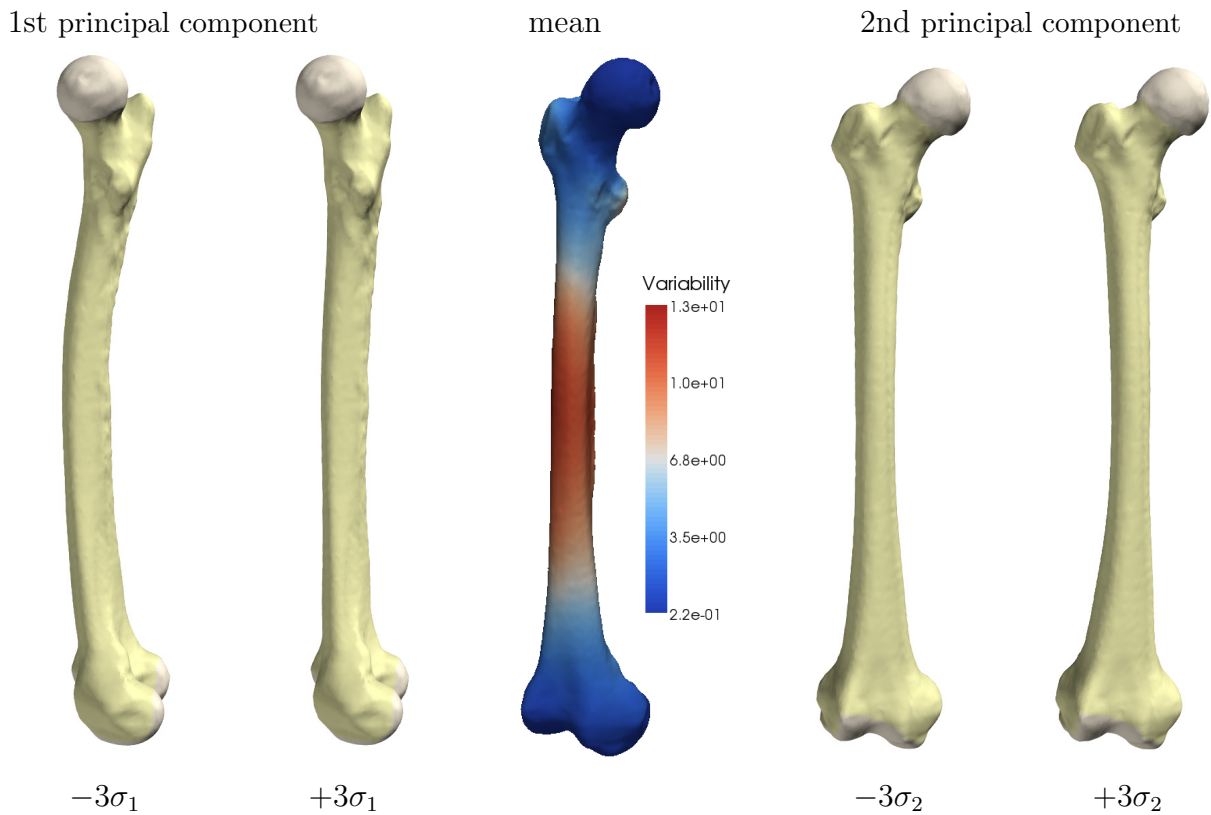


Figure 5.4: When a statistical model of the human femur bone is fitted to given joint surfaces (grey), considerable flexibility remains, visualized here by the first two principal components with standard deviation σ_1 and σ_2 . The joint surfaces are taken from the mean, seen in the middle, colored according to the remaining variability (in mm).

5.2.4 Visualizing the Flexibility

As in Section 4.2, we can use the deformations represented by the principal components to visualize the main modes of variation of the model. However here, we don't visualize the

main modes of variation of the full model, but of the model in which a part of the shape is given or fixed.

Figure 5.3 shows the first two principal components of a femur model in which the bottom (distal) half of the bone is fixed. We can see that there is a considerable variation in the shape of the proximal (top) part, especially in the angle of the head and neck of the femur. Under these deformations, the distal part stays almost, but not completely fixed. It has to be mentioned that this model is an older femur model built from only 20 examples and calculated with the method from [1], which is similar but not identical to the one described in the previous chapter.

Figure 5.4 shows a similar experiment with a newer model built from 40 examples, using the model described in the previous section. Here, we examine how strongly the joint surfaces influence the shape of the femur bone. Obviously, when the joint surfaces are given, the shaft of the bone has the largest variability. The two main modes of variation show that this variability expresses itself most prominently in the form of a bending of the shaft. On the mean bone in the middle, we display the variability of the flexibility model. We define the variability of the model at a point p as the square root of the variance at that point. With the covariance function $c(x, y)$ defined in Equation (4.6), the variability at p can be calculated as: $\sqrt{c(p, p)}$.

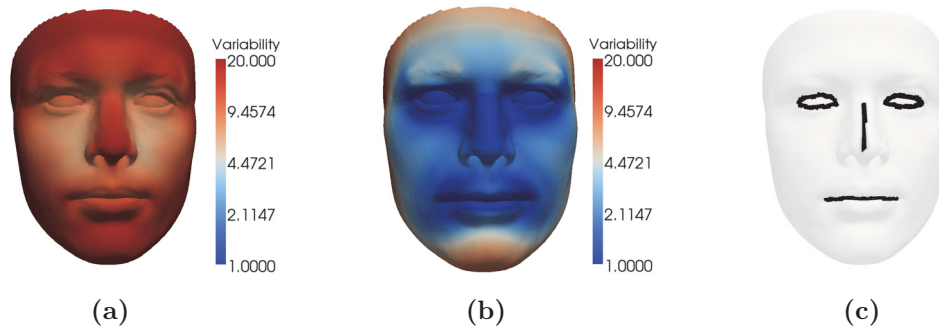


Figure 5.5: Variability of a face model. Figure (a) shows the full variability of the model. In (b), the most likely reconstruction of the sketch depicted in (c) is shown, together with the remaining variability.

In Figure 5.5, this variability is visualized for a face model, first for the regular face shape model, then for one in which eyes, nose and mouth are given in the form of a sketch similar to a smiley face. Obviously, for the full model, the variability is much higher than when the features are prescribed.

Figure 5.6 shows an example that is both entertaining and has actual clinical relevance. The statistical flexibility model can be used to predict the nose for a given face. The example we show here is an artificial example, in which we have removed the nose from the shape surface on the computer, but similar cases with real patients who have lost their nose have been treated in a project with the University Hospital Basel. A case report has been submitted for publication.

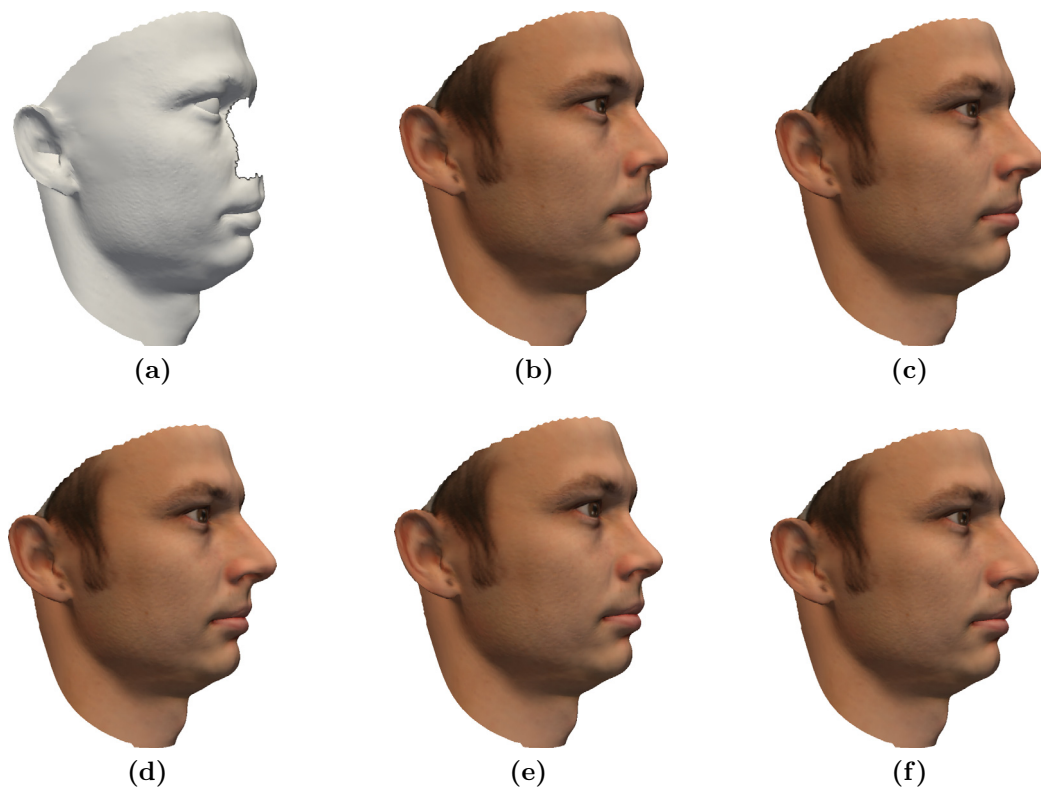


Figure 5.6: Reconstruction of a nose: (a) shows a face surface without nose. (b) shows the real face while (c) shows the reconstructed nose. In (d) and (e) we see $\pm 3\sigma$ of the main mode of variation. (f) shows a nose where the first 7 components are 3σ from the mean.

The actual real nose that belongs to this face in our example is shown in 5.6b. This face is not one of the training examples for the model. The reconstruction that our statistical model deems most probable is shown in 5.6c. Interestingly, this nose is larger than the real one. The first principal component of the flexibility model is shown in 5.6d and 5.6e. Finally, 5.6f shows an example of a very improbable but nevertheless possible nose reconstruction.

5.2.5 Limitations of the Method

In the method described above, the given points are assumed to be in correspondence with the model. When we consider them as given or fixed, we take their position in space to be fixed. For some problems however, this is too strong a constraint. For instance, if we wish to keep the contour of an object fixed, but don't know which points of the model make up the contour, we cannot use this method. We could fix some points on the contour and model the remaining flexibility as described above, but this model would not reflect the deformations that produce the same or a very similar contour but with different points.

A similar problem arises regarding the length of the bone in the femur examples in Figures 5.3 and 5.4. As long as a single point on the top and one on the bottom of the

bone is given, the length is completely determined. Therefore, with our method, we can for instance not model the shape variations of bones that have a different length but a similar shape of their joint surfaces.

This problem is even more severe than it seems at first glance: Suppose we are given a point on the bottom of the bone and one on the shaft. To apply the method described above, these points need to be in correspondence with the model, and this correspondence already determines the total length of the bone. For instance, if the point on the shaft corresponds to a point in the middle of the model, we know that the reconstructed bone will be twice as long as the distance between our two given points on the bottom and the shaft of the bone, leaving no flexibility at all. This is why, in Figures 5.3 and 5.4 all bones share the same length. In reality however, the correspondence may be unsure, especially on the shaft, which has no distinct features, and therefore, there should be some remaining flexibility in the length of the bone. Our model can currently not model this flexibility in length.

This is a conceptual problem, as our shape modeling technique relies heavily on the concept of correspondence, whereas the flexibility we may wish to model includes the possibility of uncertain or changing correspondences. Resolving this conflict will have to be the subject of future research. For now, we can only model the flexibility with respect to points that are in correspondence with our model. If this correspondence is not known a priori, it will first have to be established as described in the next section.

5.2.6 Model Fitting without known Correspondence

The fitting and partial shape modelling technique described in the previous section are only defined for target data sets \mathfrak{X} that are in correspondence with the model's reference Γ and can therefore be directly represented as a deformation of the reference with a deformation field $u \in \mathcal{H}$.

In many practical applications however, the data set \mathfrak{X} will not be in correspondence with the model. Instead it can be given in a variety of forms and representation and we need to design specific distance measures to quantify the distance between the data set and the model. More often than not, these distance measures will be nonlinear.

Nonlinear Distance Measures Unfortunately, with these distance measures, we cannot directly apply the methods from the previous section. The likelihood function $p(u | \tilde{\alpha}) = \mathcal{N}(\bar{u} + \mathcal{Q}(\tilde{\alpha}), \sigma^2 id)$ given in Equation (5.20) corresponds to a least squares distance measure $\mathcal{D}(\alpha) = \frac{1}{2} \|u - \bar{u} + \mathcal{Q}(\tilde{\alpha})\|^2$ from Equation (5.14). If the data set \mathfrak{X} is not in correspondence with the model's reference, we will see in the following that we have to use more complicated nonlinear distance measures. This typically corresponds to a likelihood function of the form $p(u | \tilde{\alpha}) = \mathcal{N}(F(\bar{u} + \mathcal{Q}(\tilde{\alpha})), \sigma^2 id)$ with some nonlinear function F . With such a likelihood function, we can no longer apply Proposition 5.2.1, which is based on linear models, and therefore do not have an explicit representation of the conditional probability of the coefficients $\tilde{\alpha}$ given the data \mathfrak{X} . Therefore, in order to use the results for shape reconstruction and flexibility modeling from Section 5.2.2, we first compute the correspondence in a separate step and can then compute the conditional distribution according to Proposition 5.2.1.

Furthermore, nonlinear distance measures in general no longer admit a closed-form and unique solution, instead we have to try to minimize them with an iterative minimization algorithm, such as gradient descent or the slightly more sophisticated L-BFGS optimizer [76]. These methods require the derivative of the distance measure, which is why we will provide the derivative of each of the following distance measures.

Fitting vs. Registration We propose to establish the correspondence between the data \mathcal{X} and the model in a separate step. In some scenarios, the best choice is to establish the correspondence with the registration algorithm. However, if we already have a statistical shape model, we can use it to define faster and more robust methods to establish the correspondence.

In fact, shape model fitting can be interpreted as a registration algorithm with a different set of basis functions. While the registration algorithm introduced in Chapter 3, tries to establish correspondence with a deformation field $u = \sum_{i=1}^N u_i \varphi_i$, where φ_i are finite element basis functions, a model fitting algorithm finds a deformation field $u = \bar{u} + \sum_{i=1}^m \alpha_i \rho_i$, where ρ_i are the orthonormal basis functions of the model space. So instead of searching the deformation field in the finite element ansatz space $V = \text{span}\{\varphi_1, \dots, \varphi_N\}$, we search for it in the model space $\mathcal{M} = \text{span}\{\rho_1, \dots, \rho_N\}$. Of course, the model space is much smaller and restrictive than V , but it is tailored for the specific object class. As such, there is less need for the elaborate distance measures and regularizers defined in Chapter 3. The restriction to the model space and the regularization based on the normal distribution on \mathcal{M} prevents most of the adverse effects the terms of the registration function from Chapter 3 are meant to penalize. Typically, one of the distance measures defined in this chapter together with the model-based regularizer defined in Section 5.2 suffices for a reasonable fitting/registration.

Fitting a Surface Model to a Surface

A common scenario is that we wish to fit the surface model to another (possibly partial) surface of the same object class, that is not yet in correspondence with the model.

This means that the model Γ_α is a surface model defined on a reference surface $\Gamma \subset \mathbb{R}^d$, and the target data \mathfrak{X} is also a surface $\Gamma_2 \subset \mathbb{R}^d$, the most efficient way to design a distance measure is by representing Γ_2 by a distance function $I : \mathbb{R}^d \rightarrow \mathbb{R}$. Then $\Gamma_2 = \{x \in \mathbb{R}^d \mid I(x) = 0\}$ and we can define the distance measure as:

$$\mathcal{D}(\alpha) = \frac{1}{2} \int_{\Gamma} \left(I(x + u_\alpha(x)) \right)^2 do(x). \quad (5.37)$$

Obviously, when $\Gamma_\alpha \subset \Gamma_2$, we have $I(x + u_\alpha(x)) \equiv 0$ for all $x \in \Gamma$, and therefore $\mathcal{D}(\alpha) = 0$, so when the model and the data coincide, the distance measure is at its minimum of 0. The larger the set $\Gamma_2 \setminus \Gamma_\alpha$ and the further away the points of Γ_α are from Γ_2 , the higher the values of $I(x + u_\alpha(x))$ and therefore of $\mathcal{D}(\alpha)$ will be. By minimizing \mathcal{D} , we try to find the model that best fits the target surface Γ_2 .

Note however, that by minimizing \mathcal{D} , we only control if $\Gamma_\alpha \subset \Gamma_2$ and not if $\Gamma_2 \subset \Gamma_\alpha$. In most scenarios this is not necessary, because both Γ_α and Γ_2 are instances of the object

class modeled by the shape model and whenever $\Gamma_\alpha \subset \Gamma_2$, we have at least $\Gamma_\alpha \approx \Gamma_2$. In other scenarios, we actually aim for the case $\Gamma_\alpha \subsetneq \Gamma_2$, i.e. that the model instance is a true subset of the data, for instance when Γ_2 is the output of an edge-detection algorithm that finds more edges/surfaces than just the actual object surface. For other scenarios, where solutions with $\Gamma_\alpha \subsetneq \Gamma_2$ actually pose a problem, a more sophisticated distance measure has to be designed.

The partial derivatives of the distance measure defined above in Equation (5.37) are given by:

$$\partial_{\alpha_j} \mathcal{D}(\alpha) = \int_{\Gamma} I(x + u_\alpha(x)) \nabla I(x + u_\alpha(x)) \cdot \partial_{\alpha_j} u_\alpha(x) do(x) \quad (5.38)$$

$$= \int_{\Gamma} I(x + u_\alpha(x)) \nabla I(x + u_\alpha(x)) \cdot \rho_j(x) do(x) \quad (5.39)$$

Fitting an Appearance Model to an Image

While the fitting of a surface model to a surface is a common problem and arises in many applications, it usually requires the segmentation of a (d -dimensional) image in order to acquire the target surface. Instead, we can try to fit the model directly to the image. Fitting a surface model to the image usually requires some kind of edge detection in the image, which makes this approach essentially equivalent to the previous approach of surface fitting.

Instead we can use the intensity model described in Section 4.3.1 to model not only the surface of the objects by a shape model but also the inside by an intensity model. Of course, this makes sense only if the intensity of the model and the target image are comparable, e.g. if they are acquired with the same type of imaging device. Suppose now that we have a combined shape and intensity model, called an *appearance model*, with the shape variations described by the deformation model u_α and the intensity variations described by intensity model $g(\beta)$. Both are defined on a reference Γ , which we assume here to be an open measurable subset of \mathbb{R}^d , e.g. the inside of the modeled object. We denote the intensity image by the function $I(x)$. Then we can define the distance measure:

$$\mathcal{D}(\alpha, \beta) = \frac{1}{2} \int_{\Gamma} \left(\underbrace{g(\beta)(x)}_{\text{model intensity}} - \underbrace{I(x + u_\alpha(x))}_{\text{image intensity at model point}} \right)^2 dx. \quad (5.40)$$

When the image contains an instance of the object class whose shape is exactly described by $\Gamma_\alpha = \{x + u(x) \mid x \in \Gamma\}$ and whose intensity is described by the function $g(\beta)$, the distance measure $\mathcal{D}[\alpha, \beta]$ is equal to zero. If the shape and/or intensity are not matched perfectly, the distance measure will take values greater than 0, depending on how good the fit is. Remember that if the reference Γ is not aligned with the object in the image, we need to introduce an additional spatial transformation \mathbf{T}_θ into the distance measure. Additionally, the regularization term $\mathcal{R}[\alpha]$ and $\mathcal{R}[\beta]$ as proposed in Equation (5.10) may be used to penalize unlikely model parameters.

The partial derivatives of this distance measure are given as:

$$\partial_{\alpha_j} \mathcal{D}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\Gamma} (g(\boldsymbol{\beta})(x) - I(x + u_{\boldsymbol{\alpha}}(x))) \nabla I(x + u_{\boldsymbol{\alpha}}(x)) \cdot \rho_j(x) dx \quad (5.41)$$

$$\partial_{\beta_j} \mathcal{D}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\Gamma} (g(\boldsymbol{\beta})(x) - I(x + u_{\boldsymbol{\alpha}}(x))) \rho_{g,j}(x) dx \quad (5.42)$$

We hope of course that the distance measure is minimized if and only if the shape and intensity models $\Gamma_{\boldsymbol{\alpha}}$ and $g(\boldsymbol{\beta})$ coincide with the target object depicted in the image $I(x)$. But obviously shape and intensity influence each other and a bad fit for one model may be compensated with the other model. Fortunately, if the modeled objects exhibit a characteristic intensity pattern, such as in the bone model, the fitting produces reasonable results.

However, because the distance measure is only evaluated on Γ , resp. $\Gamma_{\boldsymbol{\alpha}}$, which is typically the inside of the model, all image information on the outside of $\Gamma_{\boldsymbol{\alpha}}$ is not evaluated. While the characteristic intensity pattern of the modeled object usually ensures that the fitted model covers the object in the image quite well, there is usually a small gap between the real object boundary in the image and the boundary of the model. The model seems to “shrink” inside the object. A larger discrepancy between object and model would incur a higher distance measure, but this small gap at the boundary is typically not detected by this distance measure.

There are two obvious ways to address this problem: 1.) We can complement the distance measure from Equation (5.40) with a distance measure that measures the distance between object and model boundary, such as the surface-to-surface distance measure introduced in Equation (5.37). Of course this brings us back to the problem of detecting the object boundary, which may be a very difficult problem. 2.) We can complement the appearance model distance measure with a second region-based distance measure for the outside of Γ . If we have representative training examples for this region, we can simply construct a second appearance model and use a second distance measure of the form (5.40).

Unfortunately, in many cases, no useful training data is available for the outside region. For instance, our femur model is built from isolated bones, as those in Figure 5.7a, where the outside of the bone is only occupied by air. But we would like to be able to fit it to a variety of images, like images in which with soft tissue and/or other bones are present on the outside of the bones as in Figure 5.7b. But we have no representative data for all possible backgrounds. Similarly, in face modeling there is typically no representative information about the background in front of which faces are photographed.

In these cases, in which it is not possible to build a second appearance model for the outside, we propose to design a generic model for the outside. This region-based model is estimated from the target image itself and does not rely on prior example data. This method was published in [3].

5.2.7 Generic Outside Model

In this section, we show how the appearance model fitting introduced before can be complemented by a generic outside model. The outside model is derived from the Mumford-

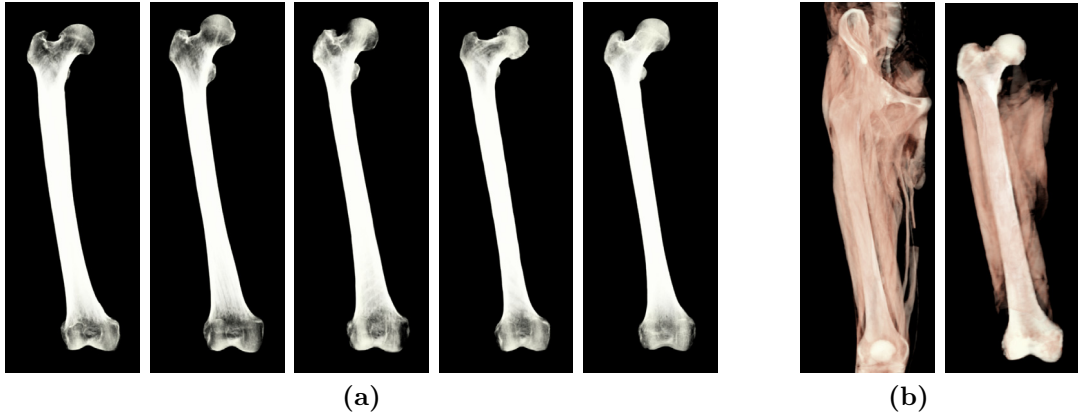


Figure 5.7: On the left: a subset of the bones used to build the appearance model. The model is built from isolated bones. No useful information is available for the outside of the bones that could be used in real segmentation tasks such as segmenting the femur in the images on the right.

Shah model for image segmentation, which we will briefly introduce next. Of course, the method could be used on any other region, but for our bone model example, it is the outside of the bones that we wish to model in this way.

5.2.8 Mumford Shah Model

In their landmark paper [53], Mumford and Shah introduced what is now known as the Mumford-Shah functional for image segmentation, which seeks to simultaneously find an edge set C and a piecewise smooth approximation J of an input image $I : \Omega \rightarrow \mathbb{R}$. In [19] Chan and Vese proposed a simplified version of this functional for the case that C is a closed contour (represented by a level set function) that separates the image domain Ω into an inside and an outside, $\text{in}(C)$ and $\text{out}(C)$ of C . In this case, the Mumford-Shah functional can be written as:

$$F(C, J) = \lambda \int_{\text{in}(C)} (J_{\text{in}} - I)^2 + \lambda \int_{\text{out}(C)} (J_{\text{out}} - I)^2 + \mu \text{length}(C) + \nu \int_{\Omega \setminus C} |\nabla J|^2, \quad (5.43)$$

where $\text{length}(C)$ denotes the length of the segment boundary C and acts as a regularization term. Typically, the functional is minimized with an interlaced algorithm. In every other iteration the boundary C is kept fixed and the image approximation J is optimized and in the next iteration J is fixed and C optimized. Mumford and Shah showed that if C is fixed, J optimizes the functional if and only if it satisfies the following elliptic boundary value problem with zero Neumann boundary conditions on each of the segments, here written out only for $\text{out}(C)$:

$$-\Delta J_{\text{out}} = \frac{\lambda}{\nu} (I - J_{\text{out}}) \text{ on } \text{out}(C) \quad \frac{\partial J_{\text{out}}}{\partial n} = 0 \text{ on } \partial(\text{out}(C)), \quad (5.44)$$

where $\frac{\partial}{\partial n}$ denotes the outer normal derivative. This means that J has to be a smoothed version of I with sharp edges on the boundary C , which is why the functional is minimal

when C coincides with edges in the image, while on the segments the image can be approximated well by smooth functions. The great advantage over methods based on actual edge detection is that even when no sharp edges are present in the image, the minimizing edge set C will still separate the different regions in the image in an optimal way when $F(C, J)$ is minimized. If $\frac{\lambda}{\nu} \rightarrow 0$, the optimal approximation J is a piecewise constant function which takes on the mean value of the function I on each of the segments, i.e. $J_{\text{out}} \equiv c_{\text{out}} = \frac{1}{|\text{out}(C)|} \int_{\text{out}(C)} I$. More sophisticated approximation strategies for $J_{\text{in,out}}$, e.g. based on texture can be found in [24].

This segmentation method separates those two regions which can be best approximated by mean intensities or smooth approximations. However, it is by no means guaranteed that these coincide with the organs we want to segment in the image.

Combining Inside and Outside Models

We now present a way to combine the prior knowledge of the statistical shape and appearance model and the generic ad-hoc modeling technique of the Mumford-Shah segmentation method. For the inside of the object, we use the appearance model exactly as described in Section 5.2.6. The outside Mumford-Shah model is derived from (5.43) with a few adjustments. Because only the outside is modeled in this way, we only use the terms concerning the outside region. Also, the length term can be omitted from the functional as the regularization properties of the statistical model provide a superior regularization method.

The terms in Equation (5.43) are defined on a part of the input image domain, whereas Equation (5.40) is defined on a part of the reference domain. To seamlessly integrate the outside terms into a common cost function, we need to transform them to the reference domain. In Equation (5.40), the spatial transformation from the reference model to the image is given by $\Phi_{\alpha}(x) := x + u_{\alpha}(x)$. Remember that if model and image are not already rigidly aligned, we need to include the additional transformation \mathbf{T}_{θ} . The relevant terms of the Chan-Vese functional can then be transformed to the domain of the reference as follows. This can also be seen as a “change of variables” from x to $\Phi_{\alpha}(x)$.

$$\begin{aligned} \lambda \int_{\text{out}(C)} (J_{\text{out}} - I)^2 + \nu \int_{\text{out}(C)} |\nabla J_{\text{out}}|^2 &= \lambda \int_{\Phi_{\alpha}(\Gamma_{\text{out}})} (J_{\text{out}} - I)^2 + \nu \int_{\Phi_{\alpha}(\Gamma_{\text{out}})} |\nabla J_{\text{out}}|^2 \\ &= \lambda \int_{\Gamma_{\text{out}}} (J_{\text{out}} \circ \Phi_{\alpha} - I \circ \Phi_{\alpha})^2 |\det D\Phi_{\alpha}| + \nu \int_{\Gamma_{\text{out}}} |\nabla J_{\text{out}} \circ \Phi_{\alpha}|^2 |\det D\Phi_{\alpha}|, \end{aligned} \quad (5.45)$$

where Γ_{out} is the outside of the model in the reference domain. In principle, Γ_{out} should be chosen so that $\Phi_{\alpha}(\Gamma_{\text{out}}) = \text{out}(C)$, but in practice, any neighborhood of Γ can be used. Then, contrary to the original integral from the Mumford-Shah functional, the transformed integral does not depend on the function or parameters we wish to optimize, which greatly simplifies the minimization. The only dependence remains in the determinant term from the transformation formula $|\det D\Phi_{\alpha}|$. However, this is where we introduce a simplifying approximation and assume $|\det D\Phi_{\alpha}| \equiv 1$, as it would be very time-consuming to compute

the derivative of the deformations caused by the matrix \mathbf{Q}_s . Secondly, this term measures the volume change caused by Φ_α and would allow the minimization of the functional simply by contracting the model to a point, which is not desirable. The combined distance measure is then given as:

$$\begin{aligned} \mathcal{D}(\alpha, \beta) = & \frac{1}{2} \int_{\Gamma} (g(\beta)(x) - I(x + u_\alpha(x)))^2 dx \\ & + \lambda \int_{\Gamma_{\text{out}}} (J_{\text{out}} \circ \Phi_\alpha - I \circ \Phi_\alpha)^2 + \nu \int_{\Gamma_{\text{out}}} |\nabla J_{\text{out}} \circ \Phi_\alpha|^2. \end{aligned} \quad (5.46)$$

The statistical deformation model u_α has been defined only for the inside model Γ in Section 5.2.6. In order to evaluate the new distance measure $\mathcal{D}(\alpha, \beta)$, it has to be extended to the outside Γ_{out} . If the training examples u_i from which the model is calculated are defined on the entire image domain of the reference, which is the case for our registration algorithm, this extension can be performed in a straight-forward manner. The mean \bar{u} is naturally extended by the mean of the fields on Γ_{out} . The eigenfunctions $\rho_i \in \mathcal{H}$ which describe the deformations of the model (see Equation (4.17)) have to be extended in a way that the outside deformation matches the inside deformation. This can be achieved by the same linear combination of the training examples on the outside as on the inside. These linear combination are given in Equation (4.36) as:

$$\rho_i = \sum_{j=1}^n \frac{1}{\sqrt{n}\sigma_i} v_{ij} (u_j - \bar{u}). \quad (5.47)$$

In terms of the matrix formulation developed in Section 4.3 for discrete models, this can be written as $\mathbf{U} = \frac{1}{\sqrt{n}} \mathbf{XVW}^{-1}$, cf. Equation 4.47.

5.2.9 Implementation

The minimization of the functional G defined in Equation (5.46) is handled in an interlaced algorithm similar to that of the Mumford-Shah method described in Section 5.2.8: We alternately calculate the ad-hoc model J_{out} for the current parameters α, β, ρ , and find the parameters for the next iteration step with a standard optimization algorithm; we use the LBFGS optimizer [76]. J_{out} needs to be calculated from the image intensities as the mean intensity or according to the elliptic equation (5.44) (on Γ_{out} instead of $\text{out}(C)$). Like the inside Γ , we represent Γ_{out} by an unstructured grid, and implemented a Gaussian smoothing with Neumann zero boundary conditions on this grid to approximate the solution of Equation (5.44). A more exact solution could be achieved by computing a finite element solution on this grid.



Figure 5.8: On the left: A CT slice with its approximation by the inside and outside model. The inside is an instance of the statistical model, while the outside is modeled as a smoothed version of the image intensities. On the right, the outside is modeled as the mean value of the outside intensities, which works best for uniform outside intensity.

5.2.10 Segmentation Results

We present a few examples of bone segmentations that show the feasibility of segmentation and model fitting with our proposed combined method and its advantages over the individual methods of level set and appearance model segmentation. As we are using a strict shape constraint, none of the segmentation results are perfect. They are only the best approximation within the space of the shape model that the optimization algorithm was able to find. We used the LBFGS algorithm with a landmark-based rigid alignment of the mean model as initialization. The method is not very sensitive to the parameters. For all experiments, we have chosen $\lambda = 1$, $\eta_s = 100$, $\eta_t = 10$. For the Gaussian smoothing of the outside model, we have used a variance that corresponds to $\nu = 300$.

Figure 5.8 shows the segmentation of CT slices to illustrate the two proposed methods for calculating the generic outside models: by computing the mean or a smooth approximation of the outside region. Only when the outside of the bone is very uniform as for instance in the case of isolated bones is the constant approximation by the mean preferable over the smooth approximation. Note that the aim of the outside model is not a perfect representation of the input image, that would of course be given by the unsmoothed image itself. The aim is to give a homogeneous representation of the outside which encourages the correct placement of the model boundary because any other placement would incur a higher cost in the functional Equation (5.46).

In Figure 5.9a we compared our segmentation method to the pure Mumford-Shah segmentation provided in the Insight Toolkit, [39, 50]. We see how our method can identify the femur in a CT image with soft tissue and other bones. In contrast, the Mumford-Shah level set segmentation finds the most prominent segment boundary in the image, that between air and everything that is not air. While this is the optimal boundary from the point of view of this segmentation method, it is not the boundary we are interested in if we wish to segment femur bones. In Figure 5.9b, we see another successful segmentation with our combined model, contrasted with a result of using only the inside appearance model. As expected, the segmentation using only the inside appearance model leaves a narrow gap around the boundary of the model.

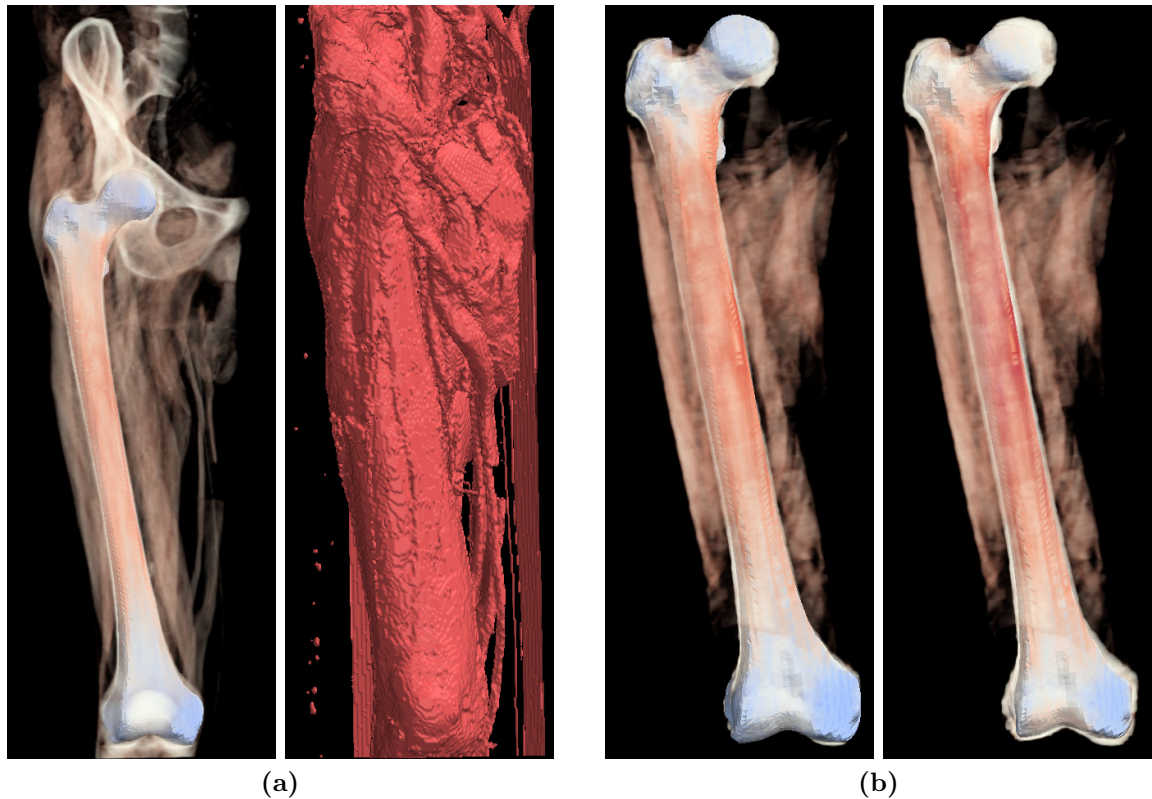


Figure 5.9: Comparison of the our segmentation method with each of the original methods. The input images are shown in Figure 5.7. On the left in (a), the proposed method identifies to femur bone, whereas the original Mumford-Shah level set segmentation on the right separates air from tissue, and therefore segmentation boundary shows the muscle tissue and not the femur bone. On the left in (b), the proposed method identifies the femur as well as the model permits, while on the right the appearance model without outside model “shrinks” and leaves a small gap between the model and the real bone surface.

5.3 Building Shape Models from Problematic Data

We have already introduced several methods to deal with problematic data, like partial shapes etc. These methods all require the use of a pre-existing shape model. In [45], we investigated a strategy to build models directly from problematic data, without the need of an existing shape model. Instead of making the registration method more robust against noise or defects as in the previous section, we employed an outlier identification method [31] to identify the problematic parts and discarded them altogether. Obviously, the data sets are then incomplete and cannot be used directly as a representative for the correct and intact anatomy of the imaged object. If we have a large set of these partial data sets however, we can still build a statistical model from a set of incomplete examples using the EM algorithm for PPCA [11]. In this algorithm, the missing part in one data set is completed using the other data sets. Obviously, this only works if the same part is not missing in the majority of examples. For the EM algorithm, no prior statistical model

is necessary. In an iterative fashion, the algorithm essentially builds a model from the completed data sets and then uses this model to compute better reconstructions for the next iteration.

For this algorithm, the data sets need to be in correspondence. Therefore, the data sets are first registered (without the statistical regularizer introduced in the previous section). The problematic parts are reflected in the registration result as unnatural deformations, which can be identified as outliers by [31] and discarded. As the problematic parts typically influence the registration result in their vicinity, they are discarded with a generous neighborhood. For our experiments with a skull model, we always discarded the entire affected anatomical structure. The complete workflow is visualized in Figure 5.10.

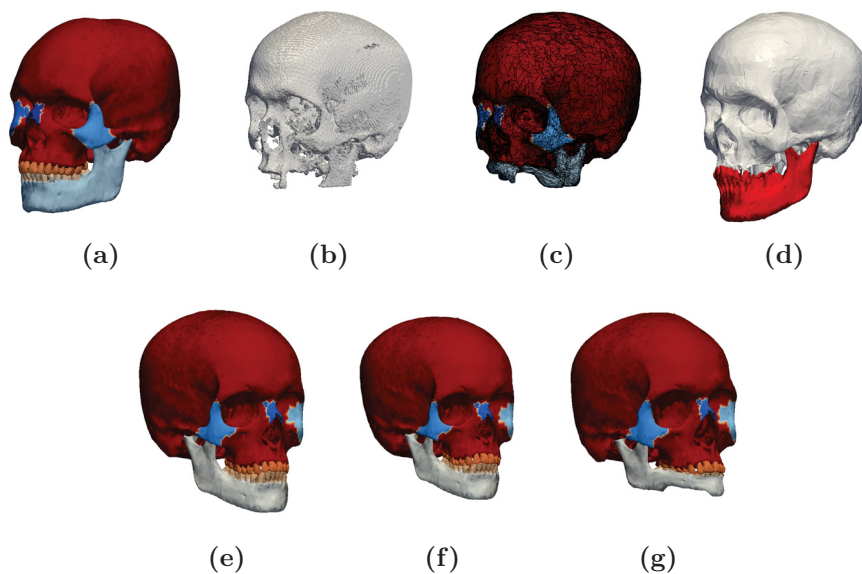


Figure 5.10: Workflow: The reference (a) is registered to a problematic data set (b). The missing parts lead to unnatural deformations in the registration and are identified as outliers. In the EM algorithm for PPCA, the outlier parts are automatically reconstructed using information from other data sets (d), and used in the construction of a statistical shape model. With mean (e) and first principal component (f). This model is not affected by the missing data, contrary to the standard PCA model without outlier detection, whose first principal component (g) shows severe artifacts caused by the missing data of just one skull.

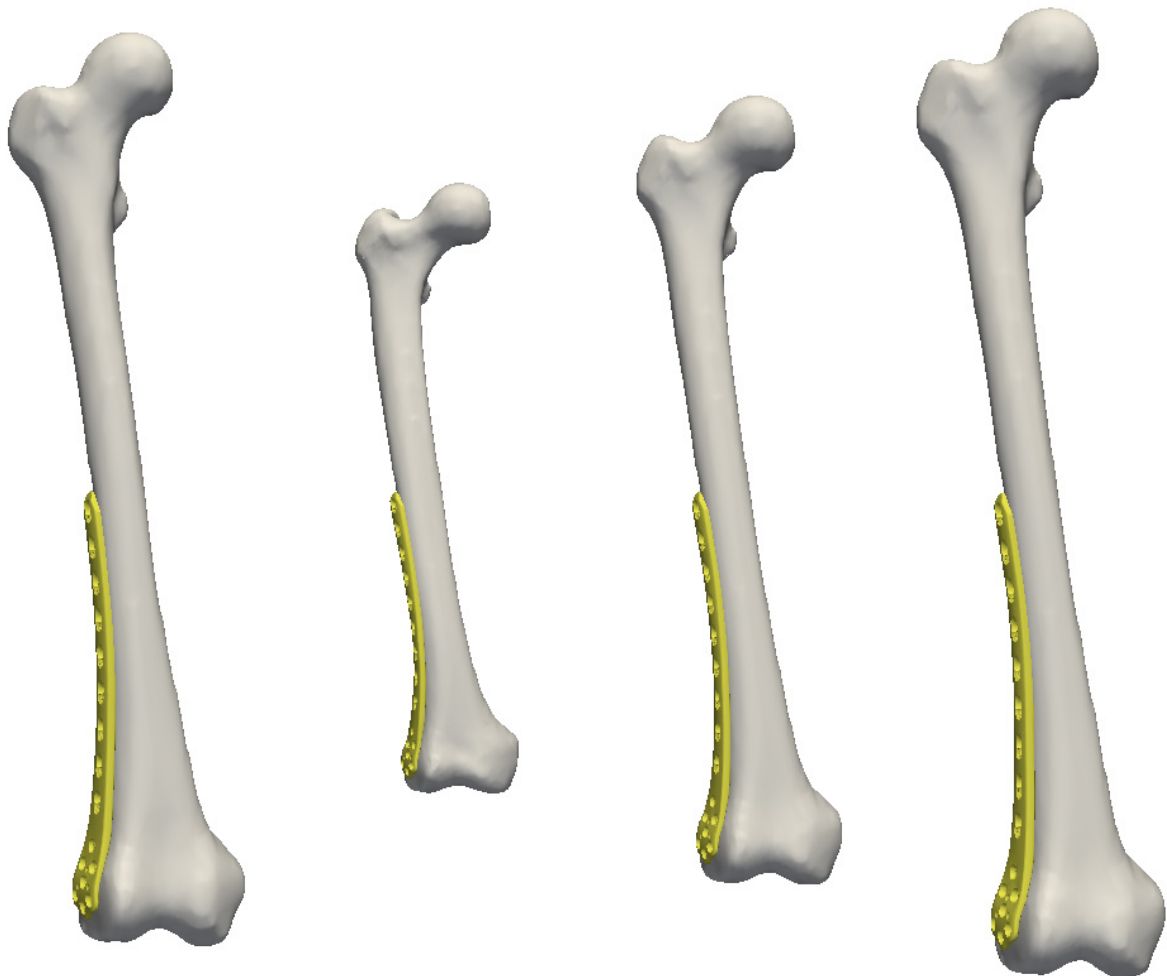
The statistical model computed by the EM algorithm, of which only the mean and first principal component are shown in Figure 5.10, is virtually unaffected by the missing data if, over all data sets, there is enough representative information for each part of the shape, even if each shape has some defect. So far, we have applied this method only to skulls, as for the other bones, we have a large collection of completely intact bones, and the few corrupted data sets can be registered using the method described in the previous sections.

5.4 Orthopedic Implant Design

In a research project with a medical device company, which is still in a preliminary stage, we have started to investigate how the statistical bone model can be used to optimize the development of implants such as plates for osteosynthesis.

Today, implants are designed based on single bones. In order to fit a wider variety of bones, an implant family is designed which varies in size and geometry, e.g. the overall curvature. These features are verified on a few other bones and feedback from doctors.

With the statistical bone model, we can give a much more detailed and comprehensive analysis of what shape variations an implant family should offer so that it covers the widest possible variety of bones.



(a) Reference bone with implant.

(b) Bones of different sizes with corresponding implant.

Figure 5.11: Designing implants that fit a variety of different bones by deforming the implant together with the bone model.

In order to get a first visual impression of the shape variations of an implant, we fitted an implant, in our example a distal femur plate, to the reference bone of our femur model, see Figure 5.11a. The implant can then be deformed together with the reference bone and we can see how the implant would have to be deformed in order to fit the deformed bone. For instance, the bones of different length seen in Figure 5.11b require implants of different length.

In a pure surface model, the corresponding deformation of the implant can be found by using the deformation of the closest points on the bone surface. In our model, the deformations can be found even more easily. Our registration algorithm introduced in Chapter 3 registers not only the surface, but also the surrounding space. The distance functions we use for registration offer a very natural extension of the surface geometry to the surrounding space. Therefore, the registration results are also available directly for the implants, and we can use them to warp not only the bone but also the attached implant.

Based on these deformation fields, we can build and visualize different models and deformations, which can all be used for implant design. First, we can visualize the main modes of variation of the bone model and simultaneously visualize the corresponding deformation of the implant. This was done for the first component of the femur model in Figure 5.11b. The corresponding implant deformations are calculated by employing the same linear combination of the registration results as in the bone model, like in Equation (5.47). It is clear that not all deformations that cause a large change in the bone also impose a large deformation of the implant. Therefore, it makes sense to calculate the main modes of the implant. This is accomplished by building a separate statistical shape model of the implant. The original implant, fitted to the reference bone, is used as the reference shape. The training deformations for this model are the deformation fields from the bone registration, restricted to the implant. In this way the implant shape model describes the deformation the implant has to undergo in order to fit the population of bones represented by the training examples. The first and main mode of variation mostly models the overall size of the implant and is very similar to the deformation pictured in Figure 5.11b.

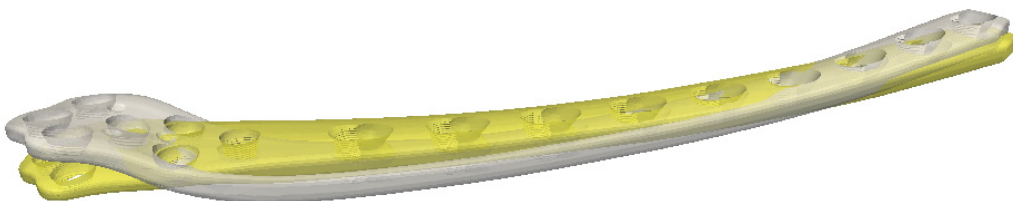


Figure 5.12: Second mode of variation of the implant model. The deformation represents ± 3 times the standard deviation σ_2 .

The second mode of variation is shown in Figure 5.12, which shows the deformations that represent ± 3 times the standard variation σ_2 of the plate model. From this we can deduce that, apart from the overall size, which is represented by the first mode of variation, the second most important shape variation that a distal femur plate family should offer is a variable overall curvature resulting in a more or less pronounced “spoon-like” end of the plate.

In order to visualize what bone shapes are associated with these implant designs, we can again employ the same linear combinations of the deformation fields, cf. Equation (5.47). Figure 5.13a shows the same plates as Figure 5.12 but with the associated bones. We see that a straighter bone with a less pronounced condyle shape calls for a straight plate while a bone with a more pronounced condyle calls for a more curved plate. In Figure 5.13b, the same analysis is performed for the third mode of variation. However, the designer may decide that it is not necessary to reproduce this mode of variation, because the mean implant would cover this bone variance well enough, simply overlaying more or less of the condyle.

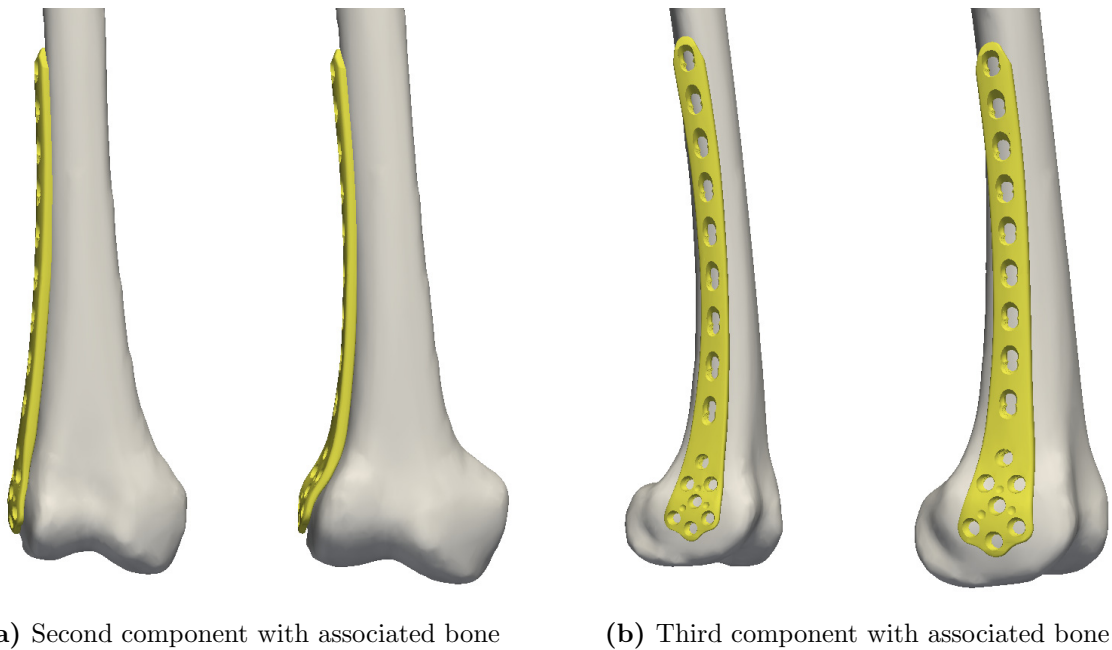


Figure 5.13: The second and third mode of variation of the implant model, together with the associated bone. This gives a visual overview of what bone variations, call for the main modes of variation of the implants.

It is still an open question how many implants with what shape variation should be used in order to best represent the variability of the implant model with as few representatives as possible. Figure 5.14 shows a scatter plot representing the coordinates of the training examples with respect to the first and second principal component. This means that each point represents the coordinates $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ of an individual training example. If we wish to design an implant family based on only the first and second principal component, we

would have to find as few representatives as possible that cover these points and/or the distribution estimated from them. At this point, it is unclear if it makes more sense to try to cover the estimated probability distribution in a systematic way or to try a clustering approach in order to cover the individual training examples optimally, e.g. with a k-nearest neighbor approach. This, and the question of how many principal components should be considered will be the object of further research.

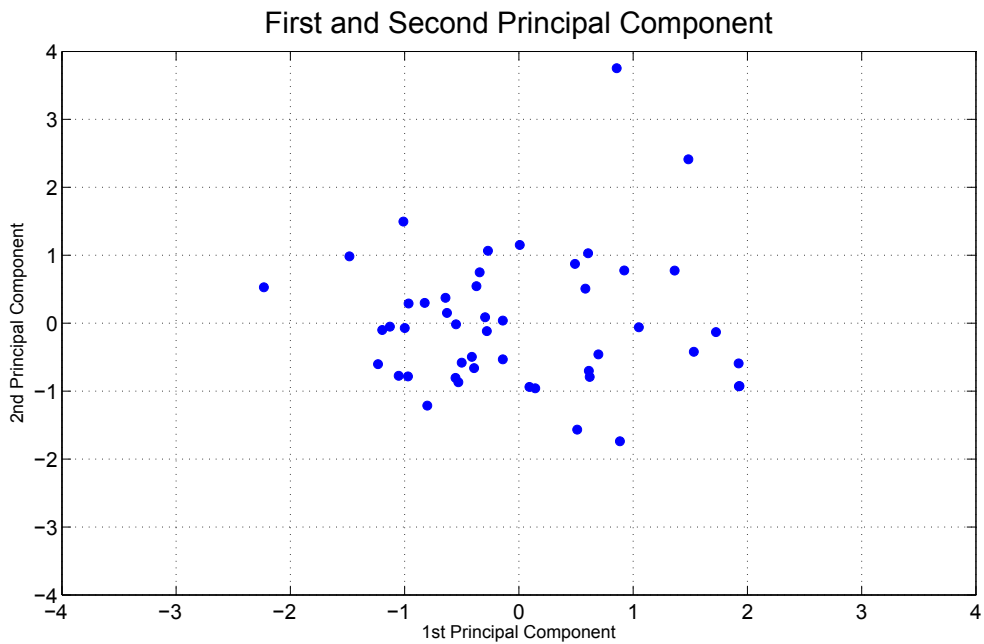


Figure 5.14: Scatter plot showing the coordinates with respect to the first two principal components of the training examples for the statistical plate shape model.

5.5 Predicting Faces from Skulls

In a similar fashion to how the statistical shape models of bone and implant are linked together in the previous section, we linked together a face and a skull model in a collaborative effort within our research group in order to predict faces from skulls and vice versa [57]. We had to deal with the additional difficulty that we had no skull data for the training examples of the face model. Therefore, the link between the different models could not be established as easily as for the implants in the previous section.

We addressed this problem by capturing a set of 20 MRI scans in addition to the example data of the skull and face models. On the MRI scans, both faces and skulls are visible and we could fit the skull and face models to these MRI scans. This provides a relationship between the skull and shape model coefficients for 20 examples, from which we estimated a linear mapping between skull and face model coefficients using ridge regression. Two results of predicting faces from skulls with this mapping can be seen in Figure 5.15. The skulls are not shown here as the differences in skull shape are so subtle that the images provide little visual information.

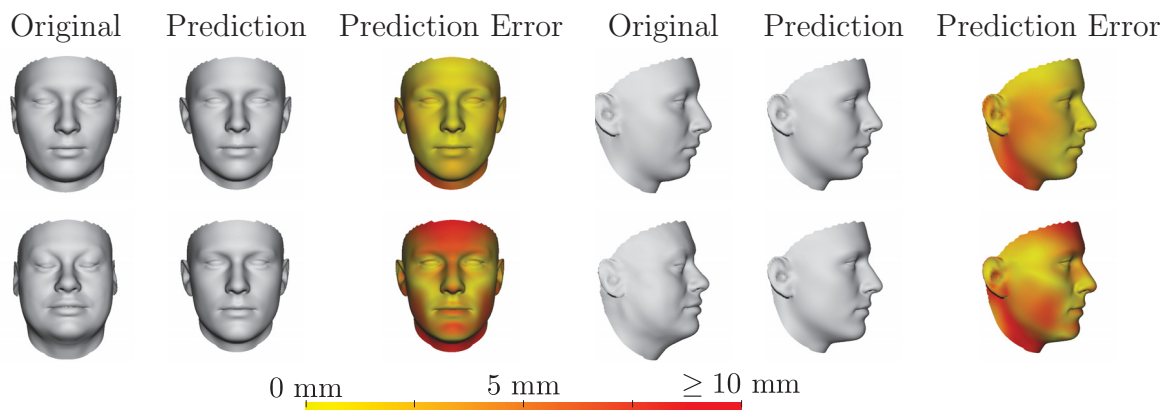


Figure 5.15: Results faces predicted from skulls. The best and worst results in terms of the Mahalanobis norm error were selected. The color-coded prediction error is the per-vertex L^2 -error orthogonal to the surface. Large errors occur at the cheeks where the soft tissue thickness depends strongly on the body weight and age.

As the body weight and age of a person is relatively independent of the skull but has a strong impact on the face, the prediction results are typically bad when a person's age and weight deviate from the norm, which can be observed in the bottom example of Figure 5.15. Therefore, we have combined the face prediction method with a method for modelling physical attributes such as age, weight, sex etc. that have an effect on the face. As the discussion of this model would take us too far into the field of face modeling, we refer the interested reader to [57] for details.

For our main application area of bone modeling, this method provides a first step towards combining related but separate bone models such as the different bones of the leg. However, so far we have only applied it to faces and skulls.

5.6 Visualizing the Density of the Subchondral Bone

One of the major research areas of the Anatomical Institute at the University of Basel is the study of the subchondral bone plate, see [46] for a comprehensive overview. The subchondral bone plate is a layer of bone that lies underneath the cartilage of each joint surface.

The main function of the subchondral bone plate is to provide support for the overlying articular cartilage. It was shown in [52] and other work reviewed in [46] that the density and mineralization of the subchondral bone plate is higher in more heavily loaded regions of the joint surface than in less loaded regions. The density and mineralization of the bone can be measured by a CT scan and it has been shown that these CT intensities have a strong correlation with the actual mechanical strength and thickness of the subchondral bone, see [51] for instance. Strength and thickness of the subchondral bone plate are in turn influenced by its functional purpose and even adapt over time to the prevalent mechanical loads. Therefore, the density distribution of the subchondral bone can be seen as an indicator for the load distribution of the joint surface. It can be used to study the typical mechanical situation of a given joint surface in a statistical fashion as well as to assess the quality of a patient's joint.

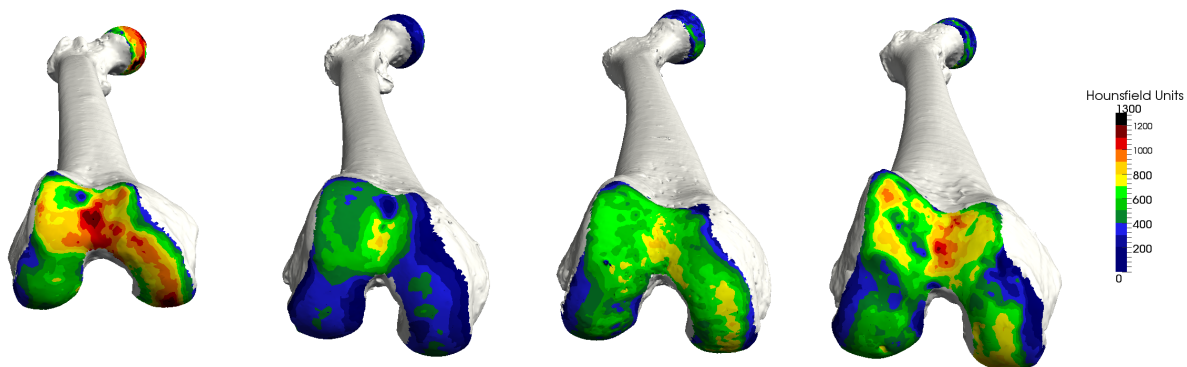


Figure 5.16: Maximum intensity projection of the subchondral bone onto the joint surfaces of four femora. Thanks to the projection in normal direction (with a depth of 4 mm) curved joint surfaces as those in the femur can be analyzed.

In order to analyze the density distribution, the subchondral bone has to be identified and its density represented visually in a concise and comprehensive way. The state of the art method is to go through the CT scan slice by slice and to identify and mark the subchondral bone plate as well the joint surface by hand. Then, the density of the subchondral bone is projected onto the joint surface by a maximum intensity projection. This means that at each point of the joint surface, the maximum density value of the underlying subchondral bone is displayed, indicating the strength of the subchondral bone plate at this point. Density distributions displayed in this way can be found in [46] and references therein. The maximum intensity projection is typically performed in one space direction. For curved joint surfaces such as the femoral condyles, multiple projections in

different directions are performed in order to approximate a projection along the normals of the joint surface.

With the shape modeling methods proposed in this thesis, we can facilitate this process in several aspects. First, if a collection of bones such as the training examples for our femur model have been brought into correspondence by registration, it suffices to identify and label the joint surface on one bone and this label can be automatically transferred in order to mark the corresponding joint surfaces on all other examples. Next, it follows from studies of the thickness of the subchondral bone, e.g. [48], and the cartilage, e.g. [65], that the thickness of the cartilage and subchondral bone plate together is less than 4 mm. Both the cartilage and the bone underneath the subchondral plate have a lower density. Therefore, projecting the maximum intensity onto the joint surface in the direction of the surface normal with a depth of 4 mm is equivalent to the projection of the manually marked subchondral bone plate. However, the model-based projection has two advantages: Firstly, it is no longer necessary to meticulously label the entire subchondral plate for each individual slice of a CT scan. It suffices to label the joint surface, which can even be performed automatically by registration. Secondly, the maximum intensity is performed orthogonal to the joint surface. This is a desirable property, which could only be approximated in the previous approach.

Figure 5.16 shows the maximum intensity projection of the subchondral bone’s density onto the joint surfaces of four femurs. The density is given in Hounsfield Units, the standard units for quantifying the radiodensity measured by the CT scanner. The projection along the surface normal allows the accurate representation of the subchondral bone density even for the highly curved joint surfaces found in the femoral head and condyles.

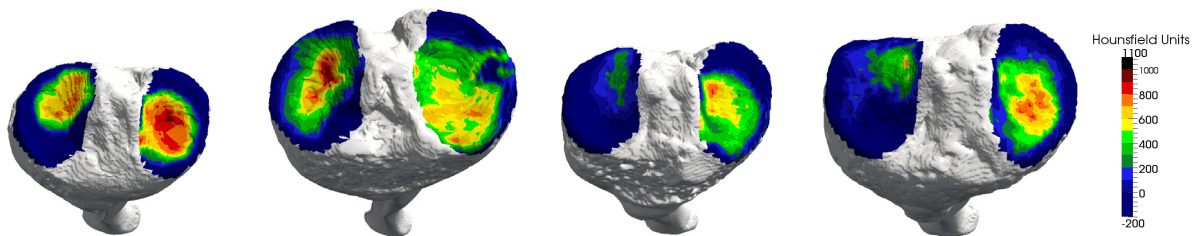


Figure 5.17: Maximum intensity projection of the subchondral bone of the tibial plateau of 4 right tibiae.

Figure 5.17 shows a similar result for four tibiae. Here, we can observe the typical density patterns of the tibia described in [46]: The first tibia shows a standard density distribution, hinting at an even load distribution. The two tibiae on the right have a considerably higher density on the medial side than on the lateral side. According to [46], this uneven density and load distribution hints at a patient with genu varum (“bow legs”). On the contrary, the second tibia from the left exhibits a higher density on the lateral than on the medial side, which is also untypical, possibly hinting at a case of genu valgum (“knock legs”).

Going further than examining such individual observations, we can build a statistical model of the subchondral bone density as described in the section on intensity models, Section 4.3.1. This allows us to visualize the mean as well as the main modes of variation

of the density. This is done in Figure 5.18 for a subchondral plate density model built from 21 tibia examples. The figure shows the first four modes of variation. The mean intensity distribution is always displayed in the middle, the density given by the mean ± 3 times the standard deviation of the mode of variation on either side. The density model shown here is calculated independently of the shape and all distributions are displayed on the shape of the tibia model's reference bone.

We see that the first mode of variation mainly models the overall density level, ranging from very dense to more delicate subchondral bone plates. We can observe that the density of the first tibia in Figure 5.17 is indeed quite close to the mean but with an overall higher density as modeled by the first mode of variation. The second mode of variation seems to represent the balance between lateral and medial condyle as discussed in the genu varum/valgum example above. This shows that, next to the overall density level, the balance between lateral and medial condyle, which reflects the load balance of the knee joint is in fact the most prominent density variation of the tibial plateau. The third and fourth modes of variation represent the variability of the position of the density maxima on the two condyles.

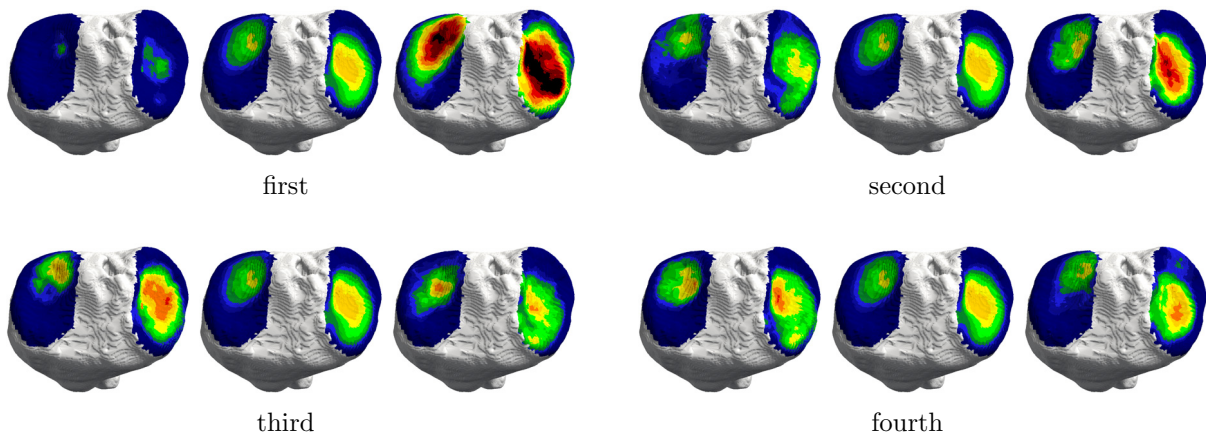


Figure 5.18: The four first principal components of the subchondral density model built from 21 tibiae. For each component, the mean is displayed in the middle, and the density $\mu \pm 3\sigma_i\rho_i$ on either side.

5.7 Fracture Reduction

An important practical application of our statistical bone model that seems to be on its way to widespread application in surgery planning is the automatic repositioning of the fragments of a broken bone. This task is referred to as *fracture reduction* in the medical domain. Research in this direction has just started and will continue until well after this thesis in a joint project with a manufacturer of surgical implants. Here, we are able to present the first results.

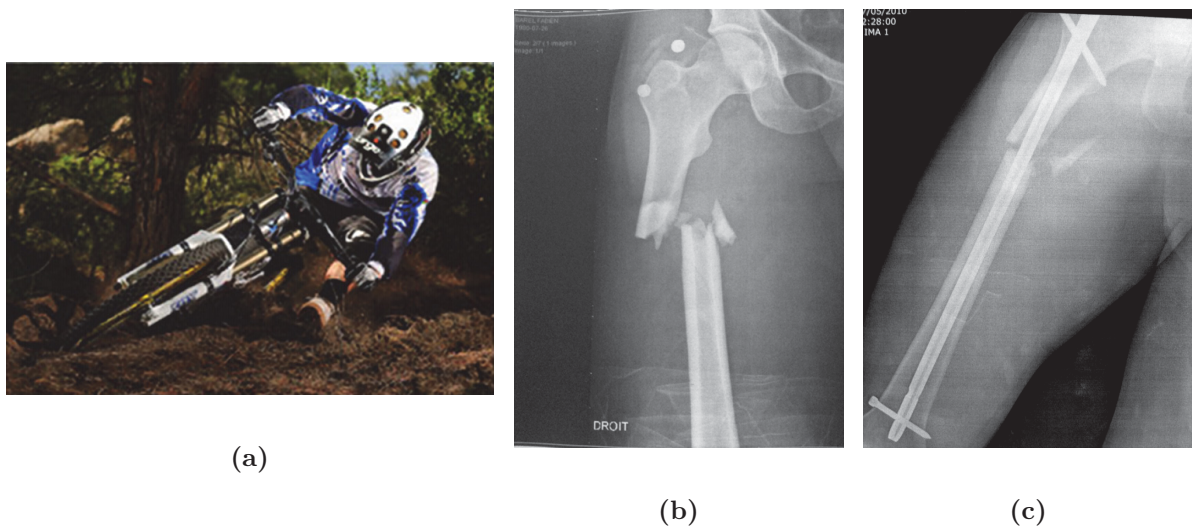


Figure 5.19: (a) shows a mountain biker who has broken his femur in a training accident. (b) and (c) show the femur before and after operation [8].

Similar to our introductory example from Chapter 1, Figure 5.19 shows a prominent mountain biker who has broken his femur this year. While it takes considerable force such as that encountered in this type of mountain biking to break such a large bone, it is not an uncommon injury. Such fractures, as well as comparable fractures in other long bones (such as tibia or humerus) are usually treated by fixing the bone with a long metal nail as in Figure 5.19c.

While this has become a fairly standard and established procedure, it remains difficult to accurately reposition the bone fragments. The metal rod is inserted through an incision near the end of the bone, and typically the fracture site is not visible directly. The only way for the operating surgeon to control the correct repositioning of the bone is by x-ray images as in Figure 5.19. These images allow a fairly accurate reposition in the longitudinal direction of the bone, even though even this alignment is not perfect in Figure 5.19c. However, rotational alignment around the axis of the bone and the nail poses a much greater challenge, as it cannot be controlled by x-ray images. In [40] Jaarsma et al. have found a rotational malalignment of over 15 degrees in 21 out of 76 patients (= 28%) for this type of operation. This means that the bottom fragment of the femur, and with it the lower leg of a patient may not be put back in its original position but instead incorrectly

rotated over 15 degrees around its axis.

Obviously, this leaves ample room for improvement. One of the companies that produce medical implants such as these femur nails is currently developing a system to improve the intraoperative control of the fracture reduction. The current position of the bone fragments is tracked in real-time by an electromagnetic tracking system and compared with an optimal reduction plan. This plan is based on a pre-operative CT image of the fractured bone. Currently, the plan is prepared manually by extracting the fragment surfaces from the CT images and aligning these surfaces by hand. The manual alignment of surfaces on a computer can be rather cumbersome and imprecise.

The application of our statistical bone model is now to automatically propose an optimal fracture reduction based on the model's inherent knowledge of the shape of intact femur bones. Given the fragments of a bone that is broken into two fragments, the model should predict how one of the fragments should be moved so that it is well aligned with the other fragment, resulting in a correctly reconstructed bone.

In principle, we are only looking for a rigid alignment of the individual fragments. We could, therefore, simply compute a rigid alignment of the fragments to an intact bone. However, this makes sense only if the intact bone has a shape that is very similar to the broken bone. This could, for instance be the contra-lateral bone, i.e. the same patient's corresponding bone from the other side. However, a CT scan of this bone may not be available for a number of reasons: because of the high additional radiation an additional CT scan would incur, because it is fractured as well, or because the two contra-lateral bones differ too much, which is actually the case for some patients.

In our current experiments, we concentrated on the reduction of the two main fragments of a fractured bone. For long bones, these can typically be described as the proximal and distal fragment, i.e. the fragment closer or further to the center of the body. Figure 5.20 shows an example of our proposed reduction method. 5.20a shows the segmented surface of an isolated bone that was broken deliberately for experimental purposes. It was CT scanned before and after fracturing. The entire bone is actually one of the example data sets for building our femur model, but it was excluded from the model for this experiment. At this stage we do not attempt to piece together the fracture as in a jigsaw puzzle. This is a different problem that may have to be tackled with a completely different method such as [74].

Instead, we wish to align the proximal and distal fragment with the help of our statistical model, in order to get an accurate reconstruction regarding the orientation and alignment of the fragments. For this purpose the model is manually cut into two fragments that represent roughly the same portion of the bone as the fragments of the broken bone, leaving out a portion to account for the middle fragment. These model fragments are shown in their initial position and shape in 5.20a. These are then fitted to the two main bone fragments with the surface fitting method described in Section 5.2.6.

The fitting consists of finding a rigid transformation for each fragment and a common set of shape parameters for both fragments together. We denote the distance function to the broken bone surface as I , the statistical model deformations by u_{α} . The two model fragments, which are transformed by transformations by \mathbf{T}_{θ} and \mathbf{T}_{ζ} , are denoted by Γ_{θ} and Γ_{ζ} . Then, the fitting can be formulated as minimizing the following distance measure with

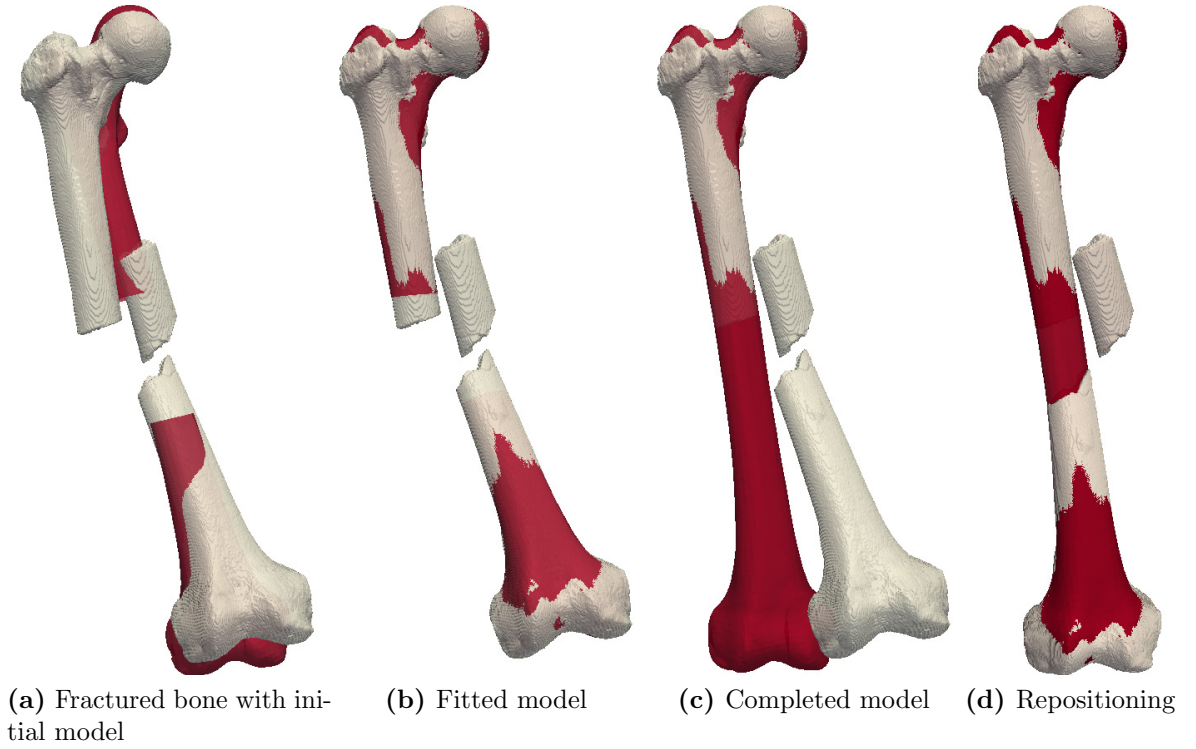


Figure 5.20: The proposed automatic fracture reduction: (a) shows a fractured femur and the initial position of two model fragments. These are then fitted to the fragments of the broken bone in (b). Presently, we fit only the two main fragments. In the fitting, we find a rigid transformation for each fragment and a common set of shape parameters. Once these are known, either model fragment can be completed to the full model (c), and the distal fragment can be repositioned to the position of the completed model (d).

respect to the shape parameters α and the two sets of rigid transformation parameters θ and ζ :

$$\mathcal{D}[\alpha, \theta, \zeta] = \frac{1}{2} \int_{\Gamma_\theta} \left(I(x + u_\alpha(x)) \right)^2 do(x) + \int_{\Gamma_\zeta} \left(I(x + u_\alpha(x)) \right)^2 do(x). \quad (5.48)$$

Once these parameters are known, either model fragment can be completed by applying the rigid transformation and shape deformation to the full model. In Figure 5.20c, the proximal fragment is completed in this way. This is the full model that best fits the position of the proximal fragment as well as the shape of both fragments. As the rigid transformation from the model to the distal fragment is known, the distal fragment can then be automatically repositioned to the correct position of within the completed model as in Figure 5.20d. This is achieved by applying the rigid transformation $\mathbf{T}_\theta \circ \mathbf{T}_\zeta^{-1}$ to the fragment, aligning it first with the reference model by \mathbf{T}_ζ^{-1} and then with the proximal fragment by \mathbf{T}_θ . As a result, we get a proposition of how, based on the model fit, a surgeon should reconstruct the full bone.

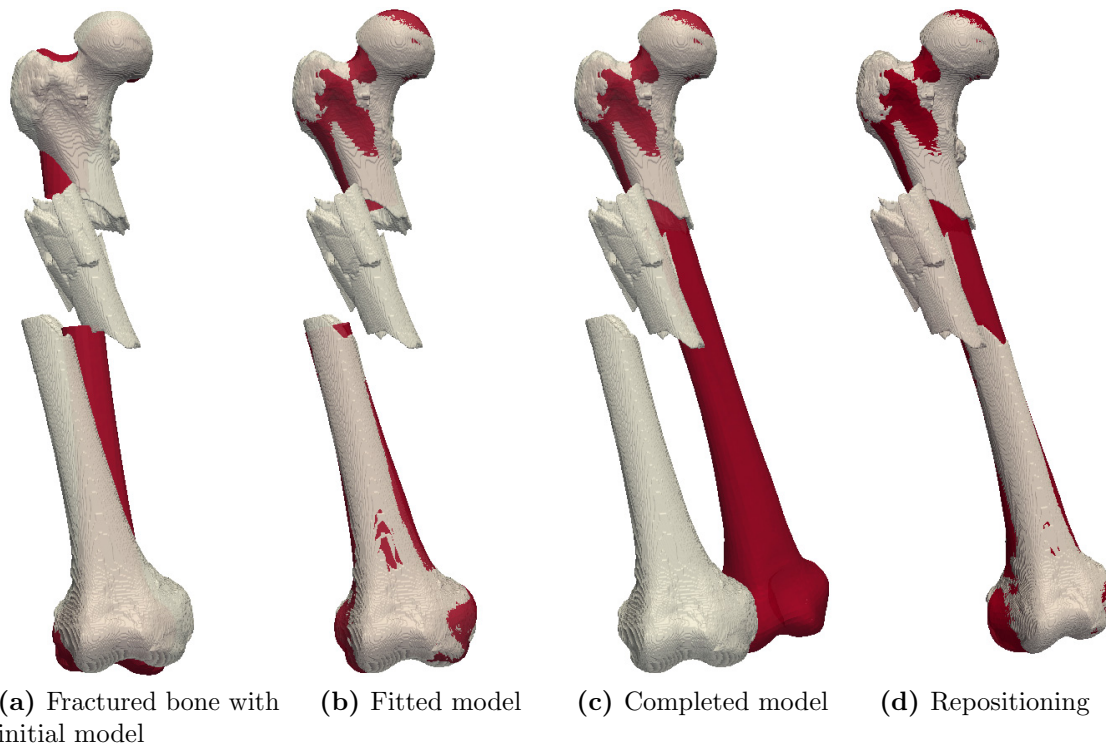


Figure 5.21: (a) shows the fractured bone with the initial model. The model is fitted to the two main fragments in (b). The proximal model is completed to a full model instance as in (c). As the rigid transforms between the model and the fragments are known, the distal fragment can be repositioned as in (d) so that it is properly aligned with the full model and the proximal fragment.

5.7.1 Open Issues

Figure 5.21 shows a second example of an automatic fracture reduction. We will use it to illustrate some open issues that will have to be addressed in future research.

Model Fragments

When comparing Figures 5.21 and 5.20, we see that the two bones are fractured in different places. Consequently, the model fragments that are fitted to the fractured bone have to be selected differently in order to roughly reflect the same portion of the bone in the model fragments as in the actual fragments. In the current implementation, we therefore have to manually cut a different set of fragments from the model for each different fracture. Obviously, it seems desirable to somehow automate this process in order to automatically and accurately cut the necessary parts from the model.

Because the proximal fragment in Figure 5.21 is rather short and contains only a small part of the bone shaft, the accuracy of the fitting, especially at the end of the fragment where it meets the shaft is not as high as we would like, see Figure 5.21c. Such inaccuracies could in principle lead to a degradation in the reduction result. On one hand it is a general

problem that smaller fragments, especially those that contain very little of the bone shaft, provide less information regarding the alignment of the bone. On the other hand, if we were able to automatically and accurately cut a portion from the model that matches the actual fragment better, the fitting result at the edge of the fragment should improve.

Bone Length

When we compare the length of the middle fragments, which we did not fit, and the space that is “reserved” for this fragment in the reconstructed bones in Figures 5.20d and 5.21d, we see that the reserved space seems to short. Indeed, the reconstructed bone seems a little too short to fit the middle fragment. This is possible because in the fitting displayed in Figures 5.20b and 5.21b, the relatively featureless shaft allows for a lengthening or shortening of the bone without a significant effect in the cost function. In principle, we would hope that an accurate fit of the prominent features on both ends of the bone, such as condyles, head and neck of the femur, trochanter etc. would determine the correct length of the bone, but obviously there is still enough room for error. In fact, while we excluded the two bones from the model in order to allow for a fair evaluation of the reduction, we do have the CT scans that were acquired prior to breaking the bones and can use these for evaluating the reconstruction of the fractured bones. The result of this evaluation can be seen in Figure 5.22. While we see that the position, orientation and shape of the bones are accurately reconstructed, we can clearly observe that the length of the intact bone is not recovered accurately. The problem is not as severe in the first case, Figure 5.22a. Here, as well as in the example from Figure 5.19 and in an actual operation, the length of the bone is completely determined by putting together the individual fragments and only the orientation and rotation of the fragments need to be determined automatically by the model. Nevertheless, such a reconstruction result is not quite satisfactory.

The problem is much more severe in the second example, Figure 5.22b, where the error in length estimation is much higher and, secondly, the fracture is so complex that a correct length estimation could actually support a surgeon in the complicated task of correctly repositioning the many small fragments of the shattered bone.

So far, it is unclear how this problem in length estimation can be reliably resolved. It is apparent from these two examples alone that the correlation between the shape of the proximal and distal ends of the femur and its length is not strong enough to gain a reliable length estimate just by fitting the proximal and distal ends. If we were able to cut exact fragments from the model that match the fractured bone’s fragments exactly up to the fracture site, the fitting result would be improved, including a better length reconstruction. The problem is however that it is not clear how to cut the correct fragments from the model, especially since the correct length of the bone is not known and is in general different from the length of the model’s reference and mean.

With a little bit of user interaction, it would most likely be possible to estimate the total length of the bone by adding together the length of the individual fragments. Then, the length of the model could be kept fixed in the fitting process, resulting in a reconstruction with a more accurate length estimation. This will most likely be the first approach we will investigate after completion of this thesis, but it would be desirable to find a more

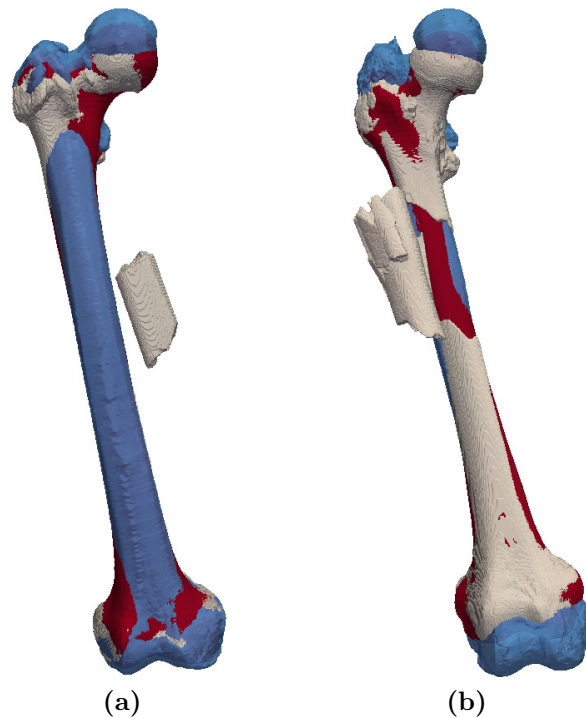


Figure 5.22: Comparing the reconstructed bones with the known ground truth surfaces, which were acquired before the bones were broken.

accurate and automatic method, hopefully in conjunction with addressing the other open questions of repositioning complicated fractures with multiple fragments possibly using the contour matching approach presented in [74] and automatically cutting appropriate fragments from the complete model.

It will also be interesting to investigate the correlation between bone shape and length. Unfortunately, this cannot readily be done with the methods for partial shape modeling proposed in Section 5.2.3, as these methods assume that the correspondence between model and target is known, whereas the inaccuracies in the length estimation stem from the fact that the correspondence in the relatively featureless shaft of the bone are not known and difficult to estimate.

In conclusion, while our experiments show that it is possible to recover the position and orientation of bone fragments for an automatic reconstruction, which should rule out erroneous rotations of over 15 degrees as reported in [40], they also raise many interesting problems for future research, going as far as posing questions about the concept of correspondence, which lies at the heart of statistical shape modeling.

6 Discussion

In this thesis, we have presented all the necessary steps for the construction and application of statistical shape models of human bones. While the broad overall modeling strategy is still the same as in the Morphable Model introduced in [13], we were able to improve every step of the process and adapt it to our application of bone modeling.

Because we can hardly expect that anybody will ever adopt the exact same framework with all the same algorithmic choices and details, we have taken great care to keep the formulation as general as possible, to separate the theoretical concepts from practical choices like discretization methods and optimization algorithms.

For our implementation of the registration method, we have chosen the local discontinuous Galerkin finite element method, because it allows us to compute the registration result efficiently on an adaptive grid and lends itself particularly well for parallelization. Other researches may prefer to calculate their registration on a regular grid, possibly using an implementation on a graphics card to speed up the calculation. By giving a general continuous formulation of the registration method the complete method, as well as its individual terms, can be used with any discretization method. In the same way, we can easily integrate additional terms into our registration functional as long as they are not specific to a different discretization method.

The lessons we have learned from our experiments certainly apply to all registration methods: The shapes should be represented by several feature images, at least one that represents its shape and one that represents its curvature. The standard regularization term that is based on controlling the first derivative of the registration result should be complemented with a volume-preservation term to achieve more natural and even distribution of the volume change that is present in inter-subject registration.

For the shape model, we have also presented a general formulation. It was already seen in [13] that the same method can be applied to model both the shape and the color of an object. By formulating the model on a general Hilbert space, we have integrated these and all other associated models in a single formulation. While this replaced the standard concepts from linear algebra which were used in [13] with their more complicated counterparts from functional analysis, we were able to show that all necessary steps for constructing a model can be formulated on general Hilbert spaces. Specifically, this formulation made it possible to include the statistical model both in the continuous formulation as well as in the discretized version of our registration algorithm, in order to make the registration more robust against outliers and missing data.

We were able to show the benefit of our model in a number of applications, some of which will hopefully see their adoption into clinical practice in the near future.

Outlook

How much closer are we to letting computers see and understand images? We were able to show that for object classes like the femur bone or the skull we can encode a kind of knowledge about the class in a computer in the form of a statistical shape model and that this model can be used to find an associated object in an image and interpret it in terms of the model parameters. Calling this “seeing” and “understanding” is a bit of an overstatement. But with this model we can perform tasks that are impossible for a computer without such prior knowledge and that are at least extremely tedious for humans without computer support. These include describing all shapes that fit a partial data set, predicting faces from skull shapes, or accurately repositioning the fragments of a broken bone.

Currently, each object class has to be modeled separately, and we don’t see the development of a universal shape model that can model all shapes at once in the future. For the application of bone modeling for medical applications, this means that a model has to be constructed for each of the bones in the human body. With new project partners we are currently acquiring additional data sets for many additional groups of bones. The next logical step is to combine these models together into a model atlas of the human skeleton. Our experiments of predicting faces from skull shapes showed how the shape parameters of two models can be linked together. But modeling complex arrangements like the knee joint, the spinal column, or the ensemble of bones in the hand will require the additional modeling of the joint articulation between different bones. Additionally, this requires a sufficient number of data sets in which several bones are visible. To be able to use more of the already available data sets, a continued research into methods to use partial and possibly damaged data is necessary.

Our shape modeling methods can be used to model other objects, for instance other organs like the heart, liver or kidney. But it is unclear how they can be applied to objects for which the concept of correspondence does not apply so easily, like vascular trees or the bronchial tree of the lungs. These can certainly be considered object classes, and correspondence can be established locally. But establishing overall correspondence like for the examples in this thesis is not possible, because the individual structure of these trees is too different from person to person. This also makes it impossible to work with common reference shapes for these types of object classes.

The strongest point of the Morphable Model is also its weakest point: correspondence. It allows the efficient modeling of shapes for which correspondence can be established but also prevents the modeling of shapes for which this is not the case. An important question for future research is how this concept can be expanded to possibly include uncertain, multiple, or only locally defined correspondence.

Even for our relatively benign examples of bone models, we have seen that our current notion of correspondence is too strict for some applications, like the modeling of length variations of the bone. This problem, along with a better estimation of the bone length for fracture reduction, will be the next topic we will try to address immediately after completion of this thesis.

Bibliography

- [1] T. Albrecht, R. Knothe, and T. Vetter. Modeling the Remaining Flexibility of Partially Fixed Statistical Shape Models. In *Workshop on the Mathematical Foundations of Computational Anatomy, MFCA'08, New York, USA*, September 2008. pages 18, 96
- [2] T. Albrecht, M. Lüthi, and T. Vetter. A statistical deformation prior for non-rigid image and shape registration. In *CVPR, IEEE Conference on*, pages 1–8, 2008. pages 18, 44, 45, 79, 84, 86
- [3] T. Albrecht and T. Vetter. Image Segmentation with a Statistical Appearance Model and a Generic Mumford-Shah Inspired Outside Model. In *Pattern Recognition*, volume 6376 of *Lecture Notes in Computer Science*, pages 61–70. Springer, 32nd DAGM Symposium edition, 2010. pages 18, 101
- [4] H.W. Alt. *Lineare Funktionalanalysis*. Springer Berlin et al., 1985. pages 31, 39, 62, 63
- [5] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002. pages 29, 48
- [6] Michel A. Audette, Frank P. Ferrie, and Terry M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4:201–217, 2000. pages 28
- [7] S.K. Balci, P. Golland, M. Shenton, and W.M. Wells. Free-form B-spline deformation model for groupwise registration. In *Medical image computing and computer-assisted intervention: MICCAI... International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 10, page 23. NIH Public Access, 2007. pages 16, 27
- [8] Fabien Barel. Update. *dirt magazine*, pages dirt.mpora.com/news/fabien-barel-update.html, dirt.mpora.com/news/fabien-barel-update-2.html, 2010. pages 116
- [9] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. II: Implementation and tests in dune. *Computing*, 82(2-3):121–138, 2008. pages 28, 49
- [10] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. I: Abstract framework. *Computing*, 82(2-3):103–119, 2008. pages 28, 49

- [11] C.M. Bishop. *Pattern recognition and machine learning*. Springer, 2006. pages 65, 70, 77, 81, 93, 106
- [12] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996. pages 32
- [13] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press, 1999. pages 12, 14, 18, 19, 57, 58, 60, 61, 62, 72, 73, 74, 77, 123
- [14] Volker Blanz and Thomas Vetter. Reconstructing the complete 3d shape of faces from partial information. *Informationstechnik und Technische Informatik*, 44(6):1–8, 2002. pages 80
- [15] Dietrich Braess. *Finite elements. Theory, fast solvers and applications in elasticity theory. 3rd corrected and enlarged ed.* Berlin: Springer. xvi, 342 p., 2003. pages 19, 29, 43, 46
- [16] C. Broit. *Optimal registration of deformed images*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1981. pages 29
- [17] R.A. Brooks. A quantitative theory of the Hounsfield unit and its application to dual energy scanning. *Journal of Computer Assisted Tomography*, 1(4):487, 1977. pages 21
- [18] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Computer Vision-ECCV 2004*, pages 25–36, 2004. pages 45
- [19] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, 2001. pages 102
- [20] GE Christensen, RD Rabbitt, and MI Miller. Deformable templates using large deformation kinematics. *Image Processing, IEEE Transactions on*, 5(10):1435–1447, 1996. pages 29
- [21] B. Cockburn and C.-W. Shu. The Local Discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463, 1998. pages 19, 29, 48
- [22] T.F. Cootes and C.J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, volume 4322, pages 236–248, 2001. pages 77
- [23] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, et al. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. pages 17

- [24] D. Cremers, M. Rousson, and R. Deriche. A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. *International Journal of Computer Vision*, 72(2):195–215, 2007. pages 103
- [25] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional. *International Journal of Computer Vision*, 50(3):295–313, 2002. pages 80
- [26] A. Dedner, R. Klöfkorn, M. Nolte, and M. Ohlberger. A generic interface for parallel and adaptive scientific computing: Abstraction principles and the DUNE-FEM module. Preprint (submitted to ACM TOMS), Mathematisches Institut, Universität Freiburg, 2009. pages 28, 49
- [27] A. Dedner, C. Rohde, B. Schupp, and M. Wesenberg. A parallel, load balanced mhd code on locally adapted, unstructured grids in 3d. *Computing and Visualization in Science*, 7:79–96, 2004. pages 28, 49
- [28] Andreas Dedner, Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. Curvature Guided Level Set Registration Using Adaptive Finite Elements. In *Pattern Recognition*, volume 4713 of *Lecture Notes in Computer Science*, pages 527–536. Springer, 29th DAGM Symposium edition, 2007. pages 17, 18, 29
- [29] M. Droske and M. Rumpf. A variational approach to non-rigid morphological registration. *SIAM Appl. Math.*, 64(2):668–687, 2004. pages 35
- [30] Lawrence C. Evans. *Partial differential equations*. Graduate Studies in Mathematics. 19. Providence, RI: American Mathematical Society (AMS). xvii, 662 p., 1998. pages 19, 38, 39, 42
- [31] P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Computational Statistics and Data Analysis*, 52(3):1694–1711, 2008. pages 106, 107
- [32] J.C. Gee and R.K. Bajcsy. Elastic matching: Continuum mechanical and probabilistic analysis. *Brain Warping*, 1998. pages 19, 85
- [33] J.C. Gee, D.R. Haynor, M. Reivich, and R. Bajcsy. Finite element approach to warping of brain images. *Medical Imaging*, pages 327–337, 1994. pages 29
- [34] S. Geman and D.E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987. pages 32
- [35] J.C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. pages 26
- [36] U. Grenander and M.I. Miller. Computational anatomy: An emerging discipline. *Quarterly of applied mathematics*, 56(4):694, 1998. pages 19, 22, 61

- [37] E. Haber and J. Modersitzki. Numerical methods for volume preserving image registration. *Inverse problems*, 20(5):1621–1638, 2004. pages 29, 36
- [38] Ian Hylands. Cam McCaul Injury Update. *Pinkbike*, pages pinkbike.com/news/cam-McCaul-Injury-update-2010.html, 2010. pages 12
- [39] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc., 2005. pages 23, 37, 49, 105
- [40] RL Jaarsma, DFM Pakvis, N. Verdonschot, J. Biert, and A. van Kampen. Rotational malalignment after intramedullary nailing of femoral fractures. *Journal of orthopaedic trauma*, 18(7):403, 2004. pages 12, 116, 121
- [41] S. Lang. *Analysis II*. Addison-Wesley, 1969. pages 40
- [42] Michael E. Leventon, W. Eric L. Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. *cvpr*, 01:1316, 2000. pages 13
- [43] Nathan Litke, Marc Droske, Martin Rumpf, and Peter Schröder. An image processing approach to surface matching. In *Symposium on Geometry Processing*, pages 207–216, 2005. pages 29
- [44] M. Lüthi, T. Albrecht, and T. Vetter. Probabilistic Modeling and Visualization of the Flexibility in Morphable Models. In *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII*, page 264. Springer, 2009. pages 18, 95
- [45] Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. Building shape models from lousy data. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 5762, pages 1–8. Springer, 2009. pages 18, 106
- [46] H. Madry, C.N. van Dijk, and M. Mueller-Gerbl. The basic science of the subchondral bone. *Knee Surgery, Sports Traumatology, Arthroscopy*, 18(4):419–433, 2010. pages 113, 114
- [47] C.R. Maurer Jr, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 265–270, 2003. pages 49
- [48] S. Milz and R. Putz. Quantitative morphology of the subchondral plate of the tibial plateau. *Journal of anatomy*, 185(Pt 1):103, 1994. pages 114
- [49] Jan Modersitzki. *Numerical Methods for Image Registration*. Oxford Science Publications, 2004. pages 19, 28, 29, 30, 35, 44
- [50] Kishore Mosaliganti, Benjamin Smith, Arnaud Gelas, Alexandre Gouaillard, and Sean Megason. Level set segmentation: Active contours without edges. *The Insight Journal*, January - June:2 – 9, 2009. pages 105

- [51] H. Mühlhofer, Y. Ercan, S. Drews, M. Matsuura, J. Meissner, U. Linsenmaier, R. Putz, and M. Müller-Gerbl. Mineralisation and mechanical strength of the subchondral bone plate of the inferior tibial facies. *Surgical and radiologic anatomy*, 31(4):237–243, 2009. pages 113
- [52] M. Müller-Gerbl, M. Dalstra, M. Ding, U. Linsenmeier, R. Putz, and I. Hvid. Distribution of strength and mineralization in the subchondral bone plate of human tibial heads. *Journal of Biomechanics*, 31(1001):123–123, 1998. pages 113
- [53] D. Mumford and J. Shah. *Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems*. Center for Intelligent Control Systems, 1988. pages 102
- [54] C. Nikou, F. Heitz, J.P. Armspach, I.J. Namer, and D. Grucker. Registration of MR/MR and MR/SPECT brain images by fast stochastic optimization of robust voxel similarity measures. *NeuroImage*, 8(1):30–43, 1998. pages 32
- [55] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. Non-rigid registration using distance functions. *Computer Vision and Image Understanding*, 89(2-3):142–165, 2003. pages 19, 29
- [56] L. Pareschi and G. Russo. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *J. Sci. Comput.*, 25(1):129–155, 2005. pages 43
- [57] P. Paysan, M. Lüthi, T. Albrecht, A. Lerch, B. Amberg, F. Santini, and T. Vetter. Face Reconstruction from Skull Shapes and Physical Attributes. In *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, page 241. Springer, 2009. pages 19, 112
- [58] S. Pieper, M. Halle, and R. Kikinis. *3D SLICER*. IEEE International Symposium on Biomedical Imaging ISBI 2004, 04 2004. pages 23, 24
- [59] J.O. Ramsay and BW Silverman. *Functional data analysis*. Springer Verlag, 2005. pages 68
- [60] T. Rohlfing, CR Maurer Jr, DA Bluemke, and MA Jacobs. Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint. *IEEE Transactions on Medical Imaging*, 22(6):730–741, 2003. pages 29
- [61] D. Rueckert, AF Frangi, and JA Schnabel. Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration. *Medical Imaging, IEEE Transactions on*, 22(8):1014–1025, 2003. pages 85
- [62] J. Schäfer and K. Strimmer. A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):32, 2005. pages 77, 78, 79, 80

- [63] B. Scholkopf, R. Herbrich, A.J. Smola, and R.C. Williamson. A generalized representer theorem. *Lecture Notes in Computer Science*, 2111:416–426, 2001. pages 73
- [64] W. Schroeder, K. Martin, and B. Lorensen. *The visualization toolkit*. Prentice-Hall Englewood Cliffs, NJ, 1996. pages 49
- [65] DET Shepherd and BB Seedhom. Thickness of human articular cartilage in joints of the lower limb. *Annals of the rheumatic diseases*, 58(1):27, 1999. pages 114
- [66] A.H. Squillacote. *The paraview guide*. Kitware, 2007. pages 49
- [67] J.-P. Thirion. Image matching as a diffusion process: an analogy with maxwell’s demons. *Medical Image Analysis*, 2(3):243–260, 1998. pages 28, 29, 32, 44
- [68] C. Thomas-Agnan. Computing a family of reproducing kernels for statistical applications. *Numerical Algorithms*, 13(1):21–32, 1996. pages 80
- [69] D. Tosun and J.L. Prince. A geometry-driven optical flow warping for spatial normalization of cortical surfaces. *Medical Imaging, IEEE Transactions on*, 27(12):1739–1753, 2008. pages 33, 34
- [70] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991. pages 13
- [71] C.J. Twining, T. Cootes, S. Marsland, V. Petrovic, R. Schestowitz, and C.J. Taylor. A unified information-theoretic approach to groupwise non-rigid registration and model building. In *Information Processing in Medical Imaging*, pages 1–14. Springer, 2005. pages 16, 27
- [72] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI*, 13:376–380, 1991. pages 26
- [73] Y. Wang and L.H. Staib. Physical model-based non-rigid registration incorporating statistical shape information. *Medical Image Analysis*, 4(1):7–20, 2000. pages 19, 85
- [74] S. Winkelbach and F.M. Wahl. Pairwise matching of 3D fragments using cluster trees. *International Journal of Computer Vision*, 78(1):1–13, 2008. pages 117, 121
- [75] Zhong Xue, Dinggang Shen, and Christos Davatzikos. Statistical representation of high-dimensional deformation fields with application to statistically constrained 3d warping. *Medical Image Analysis*, 10(5):740–751, October 2006. pages 85
- [76] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997. pages 99, 104
- [77] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003. pages 28

Curriculum Vitae

Name: Thomas Felix Albrecht

Date of Birth: March 23, 1979

Place of Birth: Mülheim an der Ruhr, Germany

Citizenship: German

Mother: Dr. med. Beate Albrecht

Father: Dr. med. Karlheinz Albrecht

Brother: Dr. med. Philipp Albrecht

Sister: Dipl. psych. Karoline Albrecht

Address: Muespacherstrasse 55
4055 Basel
Switzerland

Phone: +41 78 716 9590

E-mail: Tom.Albrecht@gmx.de

Education:

- 2006–2010:** Department of Mathematics and Computer Science,
University of Basel, Switzerland
Doctor of Philosophy in Computer Science – Grade: Magna cum Laude
Thesis: 3D Statistical Shape Models of Human Bones
their Construction using a Finite Element Registration Algorithm,
Formulation on Hilbert Spaces, and Application to Medical Image Analysis
Supervisor and Referee: Prof. Thomas Vetter, University of Basel
Co-Referee: Prof. Thomas Brox, University of Freiburg, Germany
- 1999–2006:** Faculty of Mathematics and Physics
Albert-Ludwigs University Freiburg im Breisgau, Germany
Diplom in Mathematics – Grade: “very good”
Thesis: Image Inpainting with the Mumford-Shah-Euler Image Model
Supervisor and Referee: Prof. Gerhard Dzuik, Section of Applied Mathematics,
University of Freiburg, Germany
Co-Referee: Prof. Dietmar Kröner, Applied Mathematics, University of Freiburg
- 1989–1998:** Luisenschule Essen, Germany
Abitur (German High School Diploma) – Average Grade: 1.1 “very good”
Majors: Mathematics and English

Study Abroad:

- 2002:** 2 Semesters at the Mathematical Science Institute,
Australian National University, Canberra, Australia
- 2001:** 1 Semester at the Department of Mathematics and Computer Science,
Ecole Normale Supérieure, Lyon, France
- 1995–1996:** 2 Semesters at Larkin High School, Elgin, Illinois, USA

Research and Work Experience:

- 2006–2010:** Research and Teaching Assistant,
Department of Mathematics and Computer Science, University of Basel
- 2005–2006:** Diploma Thesis: Image Inpainting with the Mumford-Shah-Euler Image
Model, Section of Applied Mathematics, University of Freiburg, Germany
- 2003–2006:** Student Assistant for Group Tutorials and Exercises,
Section of Applied Mathematics, University of Freiburg, Germany
- 2002–2003:** Kitchen Aide, Snowboard Instructor, Mountain Bike Guide
Frosch and SFI Sporttouristik, Champéry, Switzerland
- 2000–2001:** Student Assistant for Group Tutorials and Exercises,
Institute of Computer Science, University of Freiburg, Germany
- 2000:** Internship at the Institute for Medical Informatics, Biometry, and Epidemiology,
University of Essen, Germany
- 1998–1999:** Civilian Service: Nursing Assistant
Department of Orthopaedics, Alfried Krupp Hospital, Essen, Germany

Publications:

Image Segmentation with a Statistical Appearance Model and a Generic Mumford-Shah Inspired Outside Model

Thomas Albrecht and Thomas Vetter. In *Pattern Recognition: Proceedings DAGM'10: 32nd Pattern Recognition Symposium, Darmstadt, Germany*, 2010.

Building Shape Models from Lousy Data

Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009.

Probabilistic Modeling and Visualization of the Flexibility in Morphable Models

Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. In *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII*, 2009.

Face Reconstruction from Skull Shapes and Physical Attributes

Pascal Paysan, Marcel Lüthi, Thomas Albrecht, Anita Lerch, Brian Amberg, Francesco Santini, and Thomas Vetter. In *Pattern Recognition: Proceedings DAGM'09: 31st Pattern Recognition Symposium, Jena, Germany*, 2009. **Best Paper Award**

Deformable Models

Thomas Albrecht, Marcel Lüthi and Thomas Vetter In: *Encyclopedia of Biometrics*, published by Springer US, 2009

A Statistical Deformation Prior for Non-Rigid Image and Shape Registration

Thomas Albrecht, Marcel Lüthi, and Thomas Vetter. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

Modeling the Remaining Flexibility of Partially Fixed Statistical Shape Models

Thomas Albrecht, Reinhard Knothe, Thomas Vetter. In *Proceedings of the Workshop on the Mathematical Foundations of Computational Anatomy, MFCA'08, New York, USA*. 2008.

A Hierarchical, Multi-Resolution Approach for Model-Based Skull Segmentation in MRI Volumes

Marcel Lüthi, Anita Lerch, Thomas Albrecht, Zdzislaw Krol, and Thomas Vetter. *Technical Report, Computer Science Department, University of Basel*. 2008.

Sehen Lernen

Marcel Lüthi, Thomas Albrecht, and Thomas Vetter In: *Uni Nova: Das Wissenschaftsmagazin der Universität Basel*, 109, 2008

Curvature Guided Level Set Registration Using Adaptive Finite Elements

Andreas Dedner, Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. In *Proceedings of the 29th DAGM Symposium on Pattern Recognition*, 2007.

Curvature Guided Surface Registration using Level Sets

Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. In *Proceedings of CARS*, 2007

Data Mining Tools for Quality Management in Health Care

Jürgen Stausberg, Thomas Albrecht. In: *Exploratory Data Analysis in Empirical Research: Proceedings of the 25th Annual Conference of the Gesellschaft für Klassifikation eV, University of Munich*, 2001.

